Englisch · Szczepanowski

A VC–1541-es lemezegység programozása

(A nagy floppy könyv)

Lemezkezelés
Commodore 64-en kezdőknek,
haladóknak és profiknak

DATA BECKER – NOVOTRADE

Englisch · Szczepanowski

1366 B.F. →

4, 8, ... STB 255 írni

A VC–1541-es lemezegység programozása

(A nagy floppy könyv)

Lemezkezelés Commodore 64-en kezdőknek, haladóknak és profiknak

Csorgmazjő '85

DATA BECKER – NOVOTRADE

A mű eredeti címe: Das Große Floppy Buch (1983)

Fordította: DOBOSNÉ HARTYÁNYI MÁRIA
Szaklektorok: DR. OBÁDOVICS J. GYULA ÉS DR. LENGYEL JÓZSEF
A programokat lektorálta: SZERENCSI RÓBERT

A kiadásért felel: RÉNYI GÁBOR, a NOVOTRADE RT. igazgatója
Budapest, 1985
Kiadványmenedzser: BÉKÉS TAMÁS
Sorozatszerkesztő és a kötetet szerkesztette: ROCHLITZ ANDRÁS
Műszaki szerkesztő: DÉVÉNYI ERIKA
Szedte: az Alföldi Nyomda, Debrecen
Készült a Nyíregyházi Nyomdában, 20 A/5 ív terjedelemben
ISBN 963 02 3984 1

FONTOS TUDNIVALÓ

A jelen könyv keretén belül ismertetett kapcsolások, eljárások és programok nem tekintendők szabadalmi oltalom alá eső ipari termékeknek. Ezek elsősorban amatőr és oktatási célokat szolgálnak. A szerzők rendkívül nagy gondot fordítottak a kapcsolások, műszaki adatok és programok helyességére, a részletek kidolgozása során többszöri ellenőrzést végeztek. Mindez azonban nem zárja ki az esetleges hibalehetőségeket.
Az előforduló hibákért és az ebből adódó következményekért a DATA BECKER cég sem szavatosságot, sem jogi felelősséget nem vállal. Az esetlegesen előforduló hibák közlését a szerzők hálásan fogadják.

SOROZATINDÍTÓ

Miután a Commodore 64 típusú személyi számítógép a mikroszámítógépek közül Magyarországon az egyik legelterjedtebb, és így az egyik legközkedveltebb is, magától értetődik, hogy a gépek tömeges elterjedését követően a felhasználók, érdeklődők számára biztosítani kell azokat az alapvető kézikönyveket, amelyek lehetővé teszik a gépek minél sokoldalúbb kihasználását.
A DATA BECKER nyugatnémet kiadó vállalat Európában az egyik legrangosabb számítástechnikai kiadványokat megjelentető cég, mely sorozatban foglalkozik a Commodore gépekkel, a kapcsolódó perifériákkal, ill. az alkalmazott szoftverekkel.
A Commodore gépek egyik magyarországi forgalmazója, a NOVOTRADE RT. tervbe vette a DATA BECKER-könyvek egy részének, mintegy 10 kötetből álló sorozatnak a magyar nyelvű megjelentetését. Célunk a felhasználói igények jobb kiszolgálása, a mind szélesebb érdeklődés felkeltése, valamint a lehetőségek bemutatása.
A fordítás mellett szakembereink ellenőrzik a könyvekben bemutatott programok helyes működését. Ezenkívül tervezzük a könyvben szereplő programok közreadását lemezen is.
Külön hangsúlyt helyezünk arra, hogy könyveinkben magyar nyelvű, elterjedten használt (és szabványok által is javasolt) szakkifejezéseket használjunk.
A sorozat tervezett első 5 kötete a következő:

1. A VC-1541-es lemezegység programozása
2. Tippek és trükkök a C–64-en
3. A C–64 belső felépítése
4. A C–64 programozása haladóknak
5. Gépi kódú programozás

(A címek és a sorrend változhat!)
Célszerűnek látszott először ezt a kötetet közreadni, mivel a Commodore 64-et nálunk elsősorban a vállalati adminisztrációban használják kisebb adathalmazok kezelésére. Könyvünkkel a lemezkezelő műveletek mélyreható ismertetése terén mutatkozó űrt szeretnénk kitölteni.
Reméljük, hogy a könyvek nálunk is kedvelt és hasznos segédeszközei lesznek a programozási munkának.

Budapest, 1985. augusztus — Rochlitz András

ELŐSZÓ

A VC–1541 lemezegység (drive) beszerzésével a COMMODORE típusú személyi számítógépek felhasználói viszonylag olcsón jutottak hozzá egy rendkívül sokoldalú, nagyteljesítményű háttértárolóhoz. A lemezegység által kínált gazdag lehetőségek kiaknázásához azonban a szakirodalom kellő mélységű tanulmányozása is szükséges.
A könyv szerzői – Lothar Englisch és Norbert Szczepanowski – hónapokig tartó alapos munkával dolgozták fel a VC–1541-es készülék működtetésének minden titkát.
A könyv kimerítő részletességgel tárgyalja a legegyszerűbb alapismereteket – mint pl. a programok tárolása lemezen – majd fokozatosan vezeti el az Olvasót a legbonyolultabb lemezkezelési műveletek elsajátításához – mint pl. a közvetlen elérésű adathalmazok szervezése, vagy az overlay (átfedés) technika. A kezdők számára hasznos segítséget nyújtanak majd a szemléltető programok. A gyakorlott programozók viszont feltehetően kellő elégedettséggel olvassák a pontos műszaki leírásokat és örömmel tanulmányozzák a DOS (Disk Operating System) részletes dokumentációját.
A könyvet rendkívüli módon gazdagítják a futtatásra kész programok, amelyeket a felhasználónak csak be kell gépelnie és máris számos kész, használható, a lemezkezelést megkönnyítő segédprogram, illetve rutin birtokába jut. A könyv a BASIC nyelv bővítésein, a hasznos rutinokon kívül egy teljes háztartási könyvelési rendszert, illetve egy kényelmes Disk-Monitor programot is tartalmaz.

A könyv tanulmányozásához, feladatainak megoldásához és a VC–1541-es lemezegység használatához jó munkát, sőt jó szórakozást kívánunk!

Dr. Achim Becker

TARTALOMJEGYZÉK

1. FEJEZET
Bevezetés a VC–1541-es programozásába

1. FEJEZET
BEVEZETÉS A VC–1541-es PROGRAMOZÁSÁBA

1.1. Első találkozás a lemezegységgel

Az újonnan megvásárolt gyors és nagyteljesítményű háttértároló mellett az öt fejezetből álló kézikönyv terjedelme feltehetően riasztóan hat az olvasóra. Aggodalomra azonban semmi ok, a könyv szerzői ugyanis mindent megtettek annak érdekében, hogy a kezdők számára is érthető, világos magyarázatokkal, lépésről lépésre mutassák be a lemezegység programozási lehetőségeit. Meglepetéssel tapasztalják majd, hogy milyen egyszerűek a lemezkezelő programozási műveletek jó szakirodalom birtokában. Azok a kezdők, akik eddig csak programok tárolására használták a lemezegységet, megismerkedhetnek a további gazdag alkalmazási lehetőségekkel, azok a programozók pedig, akik eddig mágnesszalagos adattárolással dolgoztak, megtanulhatják és élvezhetik a lemezes adattárolás szolgáltatta lényeges előnyöket.

Annak ellenére, hogy a könyv első fejezete elsősorban a kezdők számára készült, azt javasoljuk, hogy a gyakorlott programozók is tanulmányozzák át, mert egészen biztos, hogy számukra is tudunk újat nyújtani, különösen pl. a relatív file kezelését illetően.

1.1.1. Az operációs rendszer

A lemezegységnek a VC–20-as és a COMMODORE 64-es alapgépekhez hasonlóan saját operációs rendszere van, melynek feladata a belső lemezműveletek vezérlése, illetve a géptől érkező utasítások végrehajtása.

A továbbiakban az operációs rendszer jelölésére a DOS (Disk Operating System) rövidítést fogjuk használni.

A DOS, melynek változatszáma CBM DOS V2.6 1541, természetesen kompatibilis a VC–20-as és a COMMODORE 64-es alapgépekkel. A V2.6 változat tartalmaz néhány olyan BASIC rövidítést is, amely alapkiépítésben a fent említett gépeken nem használható.

A COMMODORE 64-es és a VC–20-as gépek ugyanis a BASIC 2.0 változattal ellátva kerülnek forgalomba, a VC–1541-es azonban a BASIC 4.0 által tartalmazott, bővített lemezkezelő utasításokat is képes értelmezni, amelyek a fenti gépeken csak szimulálhatók.

A fejezet végén felsoroljuk a BASIC 2.0 összes utasítását a megfelelő BASIC 4.0 utasításokkal együtt, amelyeket a nagyobb CBM gépek (4000-es, 8000-es, 600-as és 700-as sorozat) már alapkiépítésben képesek értelmezni.

Természetesen a BASIC 4.0 bővítésként a COMMODORE 64-hez és a VC–20-hoz is csatolható, az alábbi szoftver, illetve hardver kiegészítések alkalmazásával:

VC–20:	DATA BECKER IEC-Bus és DISC BASIC
CBM–64:	DATA BECKER IEC-Bus és DISC BASIC
	SUPERTWIN
	MASTER 64.

A DATA BECKER cég kiadványa a VC–INFO tartalmazza a fenti termékek részletes leírását.

1.1.2. A mellékelt TEST/DEMO lemez

Ne zavarja a kedves felhasználót, ha kezdetben semmit sem tud kezdeni a lemezegységgel együtt megvásárolt lemezzel. A mintaprogramokon kívül ugyanis a lemez olyan segédprogramokat is tartalmaz, amelyeket bizonyos alapismeretek nélkül nem tudunk alkalmazni.
A könyv áttanulmányozása után azonban az olvasók többsége nemcsak a programok hasznos alkalmazására, hanem azok önálló megírására is képes lesz.
Célszerű tehát egyelőre félretenni a demonstrációs lemezt, amelynek részletes ismertetésére alkalmas helyen a könyvben is vissza fogunk térni.

1.1.3. A lemezek előkészítése

A boltban megvásárolt új lemezeket nem lehet előkészítés nélkül programok, illetve adatok tárolására használni. Az előkészítési folyamatot *formálás*nak nevezzük. Mit takar ez a fogalom? A különböző típusú lemezegységek más-más mechanikus elven működnek. Így legtöbb esetben a különböző típusú egységeken használt lemezek nem cserélhetők fel, azaz az egyik típusú egység nem képes a másik által „felírt" információk „elolvasására". Ennek egyik oka az, hogy a lemezek információhordozó területeinek felosztása a különböző típusú egységeken eltérő. Minden lemez *sáv*okra, és azon belül *szektor*okra van felosztva. A sávok és szektorok száma azonban az adott típusra jellemző adat. A formálás során minden szektor egy ún. *cím*et kap, amellyel a DOS a továbbiakban az adott lemezterületet azonosítja. A cím az adott sáv, illetve szektor sorszámából tevődik össze. Az egyes szektorokra ekkor azonosító információként egy 2 karakterből álló kód is felíródik, amelynek alapján a DOS felismeri, hogy a lemez formálása a VC–1541-es egységen történt. A VC–1541-es típusnál ez a kód mindig 2A. A szektor további területén 256 karakterből álló információ tárolható.
A formálás utolsó lépése a tartalomjegyzék (directory) kialakítása. A későbbi lemezhasználat során a DOS a tartalomjegyzék alapján dönti el, hogy egy szektor (vagy *blokk* – a két kifejezést a későbbiekben azonos értelemben használjuk) foglalt, vagy sem. A tartalomjegyzék mindig a lemez 18-as sávján foglal helyet.

1.1.4. A VC–1541-es néhány paramétere

Lemez:

Sávok száma:	35
Sávonkénti szektorok száma:	17–21 (a sáv mindenkori nagyságától függően)
Byte-ok (jelek) száma blokkonként:	256
Blokkok száma:	683
Szabad blokkok száma:	644 (a többit a tartalomjegyzék foglalja el)
Programok, file-ok száma lemezenként:	144

Lemezegység:
– intelligens periféria, saját processzorral és operációs rendszerrel
– csatlakozás a CBM 64-es vagy a VC–20-as soros IEC buszra; egységszám: 4–15 (alapértelmezés: 8)

1.2. A programok tárolása lemezen

A kazettás egységekhez viszonyítva a háttértárként alkalmazott lemezegység fölénye már a programok tárolásánál is megmutatkozik. A lemezes tárolás sokkal kényelmesebb, mint a szalagos. Az előbbi lényeges előnye a számítógépbe irányuló vagy a számítógépből történő átvitel sebességében mutatkozik meg. Nézzünk meg erre két példát:

Egy 3 kbyte-os program tárolásának időtartama:
- VC–1530-as kazettás egységen 75 s,
- VC–1541-es lemezegységen 12 s.

Egy 16 kbyte-os program betöltésének időtartama:
- VC–1530-as kazettás egységről 330 s,
- VC–1541-es lemezegységről 50 s.

A lemezes tárolás további előnye az áttekinthetőség. Egy program betöltéséhez elég megnézni a lemez tartalomjegyzékét, majd kiválasztani a kívánt programot. Bár a mágnesszalagon is több program tárolható, az egyes programok megkeresése rendkívül körülményes, miután a megfelelő szalagpozíció csak a szalag áttekercselésével kereshető meg.
Mielőtt hozzáfognánk a következő fejezetekben közölt példák kipróbálásához, ne feledkezzünk meg az új lemez formálásáról. Bár az 1.3.2. alfejezetben részletesen tárgyalni fogjuk, itt előzetesen bemutatjuk a lemezformálás BASIC utasítását:

```
OPEN1,8,15,"N:TESZTLEMEZ,TL": CLOSE1
```

1.2.1. A SAVE – BASIC programok tárolása

A lemezes programtárolás BASIC utasítása alig tér el a szalagos tárolásra vonatkozó utasítástól. Tegyük fel, hogy írtunk egy BASIC programot, amelyet TESZT néven mágnesszalagra akarunk rögzíteni. A megfelelő utasítás:

```
SAVE"TESZT"
```

Ha ugyanezt a programot lemezre kívánjuk rögzíteni, az eltérés mindössze annyi, hogy meg kell adnunk annak az egységnek a számát, amelyen a rögzítés történik.
Alapkiépítésben minden VC–1541-es egység a 8-as egységszámot viseli. A megfelelő utasítás tehát:

```
SAVE"TESZT",8
```

Az utasítás végrehajtása után a NEW paranccsal törölhetjük a programot a tárból, hiszen az a továbbiakban a lemezről bármikor visszatölthető.

1.2.2. A LOAD – BASIC programok betöltése

Az egységszámtól eltekintve most is azonos utasítást használhatunk a mágnesszalagról, illetve a lemezről történő programbetöltéshez.
A megfelelő utasítás kazettás egységnél:

LOAD"TESZT"

Lemezegységnél:

LOAD"TESZT",8

Az utasítás hatására az a program, amelyet TESZT néven a lemezen tároltunk, visszatöltődik a tárba.
Ha a betöltés megtörtént, a LIST paranccsal győződhetünk meg arról, hogy a program valóban a tárban van.
Ha esetleg előzőleg már volt egy program a tárban, az a betöltés során törlődik. Ennek egyszerű oka, hogy a betöltés mindig a tár egy fix helyétől (címétől) történik, amely a BASIC programok rögzített tárbeli kezdőcíme. Ennek ellenére lehetőségünk van arra is, hogy folyamatos betöltéssel egymás után több programot összefűzzünk a tárban. Az összefűzést az ún. MERGE funkció végzi, melynek használatakor a betöltés kezdőcímét meg kell változtatnunk ahhoz, hogy a már tárban lévő program ne törlődjék. A művelet részletes ismertetésére a későbbiekben kerül sor.

1.2.3. A VERIFY – a tárolt programok ellenőrzése

Nagyon ritkán, de előfordulhat, hogy egy program tárolása közben valamilyen hiba történik és így a tárolt információ nem azonos a tárbelivel. Az ilyen hibák kiküszöbölésére szolgál a VERIFY utasítás, melynek formátuma a következő:

VERIFY"PROGRAMNÉV",8

Ez az utasítás karakterenként összehasonlítja a tárban levő programot a lemezen tárolt programmal. Ha a két program azonos, a gép az OK üzenetet írja ki a képernyőre.
Írjunk néhány programsort, majd gépeljük be a következő két utasítást:

SAVE"TESZT.1",8

(A program TESZT.1. néven a lemezre íródik)

VERIFY"TESZT.1",8

(Megtörténik az ellenőrzés, azaz a tárbeli és a tárolt program összehasonlítása)
A gép szinte biztosan az OK üzenetet írja ki, ugyanis tárolási hiba valóban ritkán fordul elő. elő.

1.2.4. A SAVE "@:..." – programok felülírása

Kíséreljük meg a már lemezen lévő programunkat azonos névvel újból tárolni. A gép a FILE EXISTS hibaüzenetet küldi, a lemezegység piros jelzőlámpája (LED) villog és a parancs végrehajtása nem történik meg. A VC–1541-es nem engedi meg, hogy két programot azonos névvel tároljuk a lemezen, hiszen a programok azonosítását a DOS a név alapján végzi el. Érthető tehát a "file már létezik" hibaüzenet. Előfordulhat azonban, hogy egy programon

valamilyen módosítást végeztünk és most a javított változatot ismét szeretnénk a lemezen tárolni. Ekkor háromféle megoldással is élhetünk:

1. Az új változatot új névvel tároljuk.
2. A régi változatot töröljük a lemezről, így az új programot tárolhatjuk a régi néven.
3. A SAVE utasításban a program neve elé az "@:" jeleket írjuk (ez egy írott "a" betű, amelynek a "lába" visszakanyarodik a betű fölé).

A harmadik megoldással tulajdonképpen egy, a már lemezen meglévő program felülírására kerül sor. A DOS ezt a folyamatot a következő lépésekben végzi el:

1. Megkeresi a lemezen az első szabad blokkot és ennek címét tárolja a régi program tartalomjegyzékbeli bejegyzésében.
2. Az új programot tárolja a szabad blokkon.
3. Az új program elhelyezésének címét bejegyzi a régi helyére.
4. A régi program által lefoglalt területeket "felszabadítja".

Tehát valójában nem a régi program tényleges fizikai helyén történik az új program tárolása, hanem a lemezen lévő szabad területeken, és ezzel párhuzamosan a tartalomjegyzékben megtörténik a program elhelyezkedésére vonatkozó adminisztrációk kijavítása is. Ennek az eljárásnak alapvető következménye, hogy ha a lemezen már nincs elegendő szabad terület, a DOS a fenti utasítás végrehajtásakor hibaüzenetet küld, holott a régi program helyén az új elférne.

1.2.5. A gépi kódú programok betöltése

A gépi kódú programok a processzor elemi utasításaiból állnak. Ezek végrehajtásához a gépnek nincs szüksége az értelmező programra (interpreterre).
A gépi kódú programok tehát nem a BASIC programok fix helyére töltődnek, hanem egy ún. *abszolút tárcímre*, amely a lemezen levő program első két byte-jában van elhelyezve. A betöltés során a gépnek fel kell ismernie, hogy BASIC vagy gépi kódú programról van szó. Hogy melyik program betöltése történt meg, azt a LOAD utasításban egy ún. *másodlagos cím* megadásával jelezhetjük. A következő utasítás

```
LOAD"PROFI-MON 64",8,1
```

a PROFI-MON 64 nevű gépi kódú program betöltését végzi el, és a betöltés a lemezen levő program első két byte-jában megadott tárcímtől kezdődik, mely ez esetben decimálisan 49152. Betöltés után a gépi kódú program indítása a SYS 49152 utasítással történhet. A fenti LOAD utasításban a másodlagos cím 1. Ha egy gépi kódú programot másodlagos cím nélkül töltünk be, vagyis úgy, mint egy BASIC programot, majd megpróbáljuk RUN paranccsal elindítani, a SYNTAX ERROR IN.... (szintaktikai hiba) hibaüzenetet kapjuk. Ez természetes, hiszen az interpreter a gépi kódú programot BASIC programként próbálja értelmezni, ami nem lehetséges. Ennek megfelelően a LIST parancs hatására is program helyett egy zavaros képet kapunk.
A DOS hátrányos sajátossága, hogy a lemez tartalomjegyzékében nem különbözteti meg a gépi kódú programot a BASIC programtól. Mindkettőt a PRG file-típussal jellemzi, következésképpen a tartalomjegyzék alapján nem tudjuk eldönteni, hogy a betöltést másodlagos címmel (LOAD"...",8,1) vagy anélkül (LOAD"...",8) kell elvégeznünk. A felhasználónak magának kell megjegyeznie, hogy a lemezen milyen programokat tárol. Ha betöltés és RUN parancs után a SYNTAX ERROR üzenetet kapjuk, akkor feltehetően gépi kódú programot töltöttünk be másodlagos cím nélkül. Ekkor meg kell ismételni a betöltést, most már helyesen. A gépi kódú

program indítása nem történhet RUN paranccsal. Először meg kell határoznunk az indítási címet, amelyre többféle lehetőség van. Használhatjuk erre a célra a könyvben a későbbiekben ismertetett file-paraméter listázó programok valamelyikét. Ha megkaptuk a kezdőcímet és az történetesen 49152, akkor a programot a SYS 49152 utasítással indíthatjuk.
A kezdőcím meghatározását végzi el az alábbi néhány soros program is:

P. 1.

```
1 input"programnev: ";a$
20 open 1,8,2,"0:"+a$
30 get#1,x$:if x$="" then x$=chr$(0)
40 lb=asc(x$)
50 get#1,x$:if x$="" then x$=chr$(0)
60 hb=asc(x$)
70 close1
80 ad=hb*256+lb
90 print"start cim:";ad
```

A program a RUN parancs hatására decimálisan kiírja az adott nevű gépi program kezdőcímét.

A BASIC program működése:
A 20-as sor OPEN utasítása megnyitja a lemezen levő gépi kódú programot soros file-ként, és mivel a kezdőcím az első két byte-ban helyezkedik el, ezek tartalmát GET utasítással beolvassa. A tároláskor azért van szükség két byte-ra, mert a tárcím 255-nél nagyobb szám is lehet, az egy byte-ban tárolható legnagyobb számérték pedig 255. A két byte külön nevet visel, az első az ún. alsó byte (LOW-byte), a második az ún. felső byte (HIGH-byte). Ha a cím pl. 300, akkor az LB=44, a HB=1. (HB=INT (CIM/256), LB=CIM−256*HB, tehát HB a cím 256-tal történő osztás hányadosának egész része, LB pedig a maradék.)
A BASIC program tehát beolvassa a lemezről az alsó, majd a felső byte-ot és ezekből előállítja a tényleges decimális címet. Ha az utasítássorozat működése az Olvasó előtt a magyarázat ellenére nem lenne világos, a további fejezetekben levő részletes leírás során biztosan lehetősége lesz a megértésre.

1.2.6. A gépi kódú programok tárolása

A monitor- vagy assembler segédprogramok alatt megírt gépi kódú programok tárolása is ugyanezekkel a segédprogramokkal történik. A gépi kódú programok kezelését azonban egyszerű BASIC programmal is elvégezhetjük. A gépi utasításoknak megfelelő decimális kódokat DATA sorokba kell rögzíteni, majd ezeket beolvasás után POKE utasítással kell elhelyezni a megfelelő tárcímekre.
Az alábbi BASIC program futtatásával betöltünk a tárba egy gépi kódú programot, a 10-es sorban megadott kezdőcímtől a 20-as sorban megadott címig terjedően:

```
10 AA=KEZDŐCÍM
20 EA=VÉGCÍM
30 FOR I=AA TO EA
40 READ X
50 POKE I,PEEK(X)
```

```
60 NEXT I
80 DATA ........................
90 DATA ........................
```

A kezdőcímet és a végcímet szintén decimális értékben kell megadni, mivel a POKE utasítás argumentuma decimális.
Ennek a betöltési módnak hátránya, hogy minden betöltés egy BASIC program futtatásával jár együtt, amelyben a READ-DATA végrehajtása meglehetősen lassú. A tényleges gépi kódú rutin indításához ismernünk kell a kezdőcímet, hiszen ez a SYS utasításhoz szükséges.
Gyorsabb és elegánsabb megoldás egy olyan betöltő program használata, amely a lemezről közvetlenül a gépi kódú programnak megfelelő karaktereket olvassa be, majd a POKE utasítással az azoknak megfelelő ASCII kódokat írja be a kiválasztott tárcímre. Ahhoz azonban, hogy ezt a betöltési módot használhassuk, a gépi kódú programot utasításonként karakteres formában kell a lemezre kiírni. Ezt a feladatot végzi el az alábbi program:

```
10 AA=KEZDŐCÍM
20 EA=VÉGCÍM
30 OPEN 1,8,1"PROGRAMNÉV"
40 HB=INT (AA/256) : LB=AA–HB*256
50 PRINT#1,CHR$(LB);CHR$(HB);
60 FOR I=AA TO EA
70 PRINT#1,CHR$(PEEK(I));
80 NEXT I
90 CLOSE 1
```

A 10-es és 20-as sorokban itt is meg kell adni a tár kezdő- illetve végcímét.
Természetesen egyszer most is le kell futtatnunk az előző BASIC programot ahhoz, hogy a gépi kódú program tényleg az adott tárterületre kerüljön.
Ha a két feladatot (a gépi kódú program tárbeli elhelyezését és annak karakteres formában történő lemezre írását) egy lépésben akarjuk elvégezni, akkor a következő BASIC programmal dolgozhatunk:

```
10 AA=KEZDŐCÍM
20 EA=VÉGCÍM
30 OPEN 1,8,1, "PROGRAMNÉV"
40 HB=INT(AA/256):LB=AA–HB*256
50 PRINT#1,CHR$(LB);CHR$(HB);
60 FOR I=AA TO EA
70 READ X
80 PRINT#1,CHR$(X);
90 NEXT I
100 CLOSE 1
110 DATA ......................
120 DATA ......................
```

A program egyszeri lefuttatása után a gépi kódú programot a LOAD "PROGRAMNÉV",8,1 utasítással (és nem a BASIC programmal!) bármikor betölthetjük a lemezről arra a tárterületre, ahonnan kivittük. Az indítást most is a SYS utasítással kell elvégezni, amelyhez ismerni kell a tárbeli kezdőcímet. Ezt azonban egy betöltő programmal könnyen kiküszöbölhetjük:

```
10 IF A=0 THEN A=1:LOAD"PROGRAMNÉV", 8,1
20 SYS (Kezdőcím)
```

A 10-es sorban elhelyezett IF utasításra feltétlenül szükség van. Próbáljuk meg IF utasítás nélkül elvégezni a betöltést:

```
10 LOAD"PROGRAMNÉV",8,1
20 SYS (Kezdőcím)
```

Tapasztalni fogjuk, hogy a 20-as utasításra sohasem adódik át a vezérlés. Ha ugyanis a LOAD utasítást nem parancsként, hanem programutasításként használjuk gépi kódú program betöltésére, a betöltés után a vezérlést ismét az első BASIC sorra adja a rendszer. Az előbbi megoldásban, bár a vezérlés betöltés után ismét a 10-es sorra adódik, az A változó értéke már nem nulla, tehát a LOAD-ot a program másodszor nem hajtja végre.
A rövid betöltő programok használatának előnye, hogy nem kell fejből ismerni a gépi kódú program indítási címét.

1.3. A lemezegység rendszerutasításai

Amint azt a korábbiakban már említettük, a VC–1541-es egység a nagyobb kategóriájú CBM lemezegységekhez hasonlóan saját processzorral és operációs rendszerrel rendelkező intelligens periféria.
Az operációs rendszer, a DOS nem foglal el külön tárterületet az alapgép központi tárából, ami nem megszokott dolog, hiszen a legtöbb periféria egy, a központi tárba betöltendő operációs rendszerrel dolgozik. További érdekessége és ugyanakkor előnyös tulajdonsága a DOS-nak, hogy a lemezműveleteket teljesen önállóan, az alapgép igénybevétele nélkül hajtja végre. Minthogy azonban így a lemezműveletekre vonatkozó utasítások nem szerepelnek a COMMODORE 64-es és a VC–20-as utasításkészletében, ezeket az utasításokat valamilyen módon el kell juttatni a lemezegységhez. Átvitel után ezek az utasítások meghívják a kívánt feladatot ténylegesen elvégző DOS rutinokat.

1.3.1. Utasításátvitel a lemezegységhez

Minden utasítást vagy parancsot egy ún. *csatornán* keresztül küldünk el a lemezegységhez. A rendelkezésre álló 16 csatorna közül a 15-ös a parancscsatorna (vagy utasításcsatorna). Az adatátvitel a parancscsatornán a következőképpen történik:

- a csatorna megnyitása (OPEN)
- adatátvitel (PRINT)
- a csatorna lezárása (CLOSE)

Az OPEN utasításban mindig meg kell adni annak az egységnek a fizikai számát, amelyhez a parancsot küldjük, továbbá egy ún. logikai file számot, amelyhez a parancscsatornát hozzárendeltük.

```
OPEN logikai file szám, 8,15,"utasítás"
```

Ebben az utasításban a címzett egység a 8-as, a parancscsatorna a 15-ös, a logikai file szám értéke 1–127 közé eső tetszőleges egész szám.

A logikai file szám (lfn) az egységre vonatkozó átviteli utasítások (PRINT#, INPUT#, GET#) egyértelmű hozzárendelését szolgálja. A tényleges rendszerutasításokat vagy közvetlenül az OPEN után kell elhelyezni, vagy egy külön PRINT utasításban, a megnyitást (OPEN) követően. A parancscsatorna lezárásáig tetszőleges számú rendszerutasítást kiadhatunk, de természetesen a PRINT után mindig a megfelelő logikai file számot kell beírni.

1.3.2. A NEW – a lemez formálása

A korábbiakban már szó esett az új lemezek előkészítéséről, vagy más néven a formálásról. A formálást a NEW utasítással végezhetjük el, melyet a kezdőbetűjével (N) rövidítve is használhatunk.
Az utasítás formátuma:

```
NEW: lemeznév, azonosító
```

(Az azonosító helyett gyakran használjuk az "ID" – identifier = azonosító, angol szóból eredő rövidítést.)
A lemeznév maximum 16 karakterből állhat és bekerül a lemez tartalomjegyzékének fejlécébe. Az azonosító (ID) két tetszőleges karakter, amelynek alapján a DOS felismeri, ha a lemezegységben munka közben lemezcsere történt. Azáltal, hogy az ID tetszőlegesen választható, lehetővé válik, hogy a felhasználó a saját elképzelése szerint valamilyen rendszert alakítson ki a lemezei között. (Pl., ha 99-nél nincs több lemeze, célszerű a lemezek egyszerű sorszámozása.)
Egy példa a lemez formálására:

```
OPEN1,8,15,"NEW:TESZTLEMEZ,TL"
```

Tegyünk egy lemezt a 8-as egységbe, gépeljük be a fenti utasítást, majd nyomjuk meg a RETURN billentyűt.
Az egység sajátos kerregő hangot adva megkezdi a lemez formálását, amely kb. 80 másodpercig tart. Formálás közben a gép szabadon használható, hiszen a lemezműveletet az egység saját processzora végzi. Az utasítás rövidített alakja:

```
OPEN 1,8,15,"N:TESZTLEMEZ,TL"
```

Az alábbi két utasítás együtt egyenértékű a fentivel:

```
OPEN 1,8,15              (a parancscsatorna megnyitása)
PRINT#1,"N:TESZTLEMEZ,TL"
```

Megnyitás után mindkét megoldásnál tetszőleges számú további rendszerutasítást adhatunk ki mindaddig, amíg a parancscsatornához rendelt logikai file-t le nem zártuk (CLOSE 1). A PRINT utasításokban azonban következetesen az OPEN-ben szereplő logikai file számot kell használnunk.

1.3.3. A hibacsatorna leolvasása

A programozási munka során időnként előfordul, hogy a program nem működik megfelelően, vagy legalábbis nem úgy működik, ahogyan a programozó szeretné. A hibák „kellemesebb" fajtáját a BASIC interpreter vagy az operációs rendszer felismeri és annak okáról hibaüzenetet küld a programozónak.
A lemezműveletek közben fellépő hiba esetén általában villog az egységen a piros lámpa, a hiba jellegéről azonban az alapgép operációs rendszere nem tud tájékoztatni, hiszen ezeket a műveleteket a lemezegység processzora végzi.
Ahhoz, hogy a hiba okát megállapíthassuk, a 15-ös parancscsatornán keresztül INPUT utasítással le kell olvasnunk a lemezegység által küldött hibaüzenetet.
Minden hibaüzenet négy mezőből áll:

1. a hiba azonosító száma (numerikus)
2. a hiba szöveges megnevezése (karakteres)
3. a hiba előfordulási helyének sávja (numerikus)
4. a hiba előfordulási helyének szektora (numerikus)

A sáv és szektor meghatározza a hiba helyét. A hibaüzenetnek ezt a 4 mezőjét 4 változóba kell beolvasni, ahol a második kötelezően füzér (string). Tehát az INPUT utasítást 4 változónak kell követnie.
Példa a hibacsatorna leolvasására:

```
OPEN 1,8,15                                (ha még nem történt meg)
INPUT#1,FN,FB$,SP,SE
CLOSE 1
```

Miután azonban az INPUT utasítás parancsként nem használható, a hibát a programon belül kell leolvasni. Ez gyakorlatilag annyit jelent, hogy a fenti utasításokat sorszámmal kell ellátni, s a programot RUN-nal indítani:

P. 2.

```
10 open1,8,15
20 input#1,hs,hm$,sa,se
30 prinths;hm$;sa;se:    rem kiiras a kepernyore
40 close1
```

A program kipróbálásához idézzünk elő egy lemezhibát:

```
OPEN 1,8,15, "NEW TESZTLEMEZ,T1"
CLOSE 1
```

Ha végre akarjuk hajtani ezeket az utasításokat, a vörös színű LED villog az egységen. Felismertük a hibát?
Hiányzik a kettőspont a NEW utasítás után. Gépeljük be most a hibacsatorna leolvasásához szükséges utasítássorozatot és indítsunk RUN-nal. Ekkor a képernyőn az alábbi üzenet jelenik meg:

```
34 SYNTAX ERROR 0 0
```

A 34 a hiba száma, ezt követi a szöveges hibaüzenet. A sáv és a szektor mező 0, mert ez a hiba nem igényli ezeket az adatokat. Ha hiba észlelése nélkül olvassuk le a hibacsatornát, a

```
0 OK 0 0
```

üzenetet kapjuk.

Ha a lemezegység használata közben a vörös színű LED villog, mindenekelőtt az általunk írt utasításokat ellenőrizzük, mert így a hiba többnyire könnyen felismerhető, mint ezt a fenti példában is láthattuk. Az 1.6. fejezet tartalmazza az összes hibaüzenetet és a hibák okainak részletes leírását.

1.3.4. A LOAD "$",8 – a tartalomjegyzék betöltése

A DOS a tartalomjegyzékben katalogizálja a lemezen tárolt összes file-t (a programokat és az adathalmazokat).

Sohase feledkezzünk meg arról, hogy a tartalomjegyzék betöltésekor elvész az a program, amely előzőleg a tárban volt. A tartalomjegyzéket a

```
LOAD"$",8
```

parancs végrehajtásával lehet betölteni és a LIST paranccsal a képernyőre íratni.

Vegyük elő az egységhez mellékelt TEST/DEMO lemezt és töltsük be a tartalomjegyzéket a fenti paranccsal. Ekkor a képernyőn az alábbiak jelennek meg:

```
0 ▮"1541TEST/DEMO     " ZX 2A
13    "HOW TO USE"        PRG
5     "HOW PART TWO"      PRG
4     "VIC-20 WEDGE"      PRG
1     "C-64 WEDGE"        PRG
4     "DOS 5.1"           PRG
11    "COPY/ALL"          PRG
9     "PRINTER TEST"      PRG
4     "DISK ADDR CHANGE"  PRG
4     "DIR"               PRG
6     "VIEW BAM"          PRG
4     "CHECK DISK"        PRG
14    "DISPLAY T&S"       PRG
9     "PERFORMANCE TEST"  PRG
5     "SEQUENTIAL FILE"   PRG
13    "RANDOM FIAL"       PRG
23    "CBM-64 DEMO"       PRG
8     "LEMEZFOGLALTSAG"   PRG
65    "BETETPROGRAM"      PRG
5     "LEMEZJAVITAS"      PRG
440 BLOCKS FREE.
```

(Elképzelhető, hogy az olvasó lemezének tartalomjegyzéke eltér attól, amit mi közlünk, ugyanis időközönként a COMMODORE cég kisebb módosításokat végez a DEMO lemezen.)

Vizsgáljuk meg részleteiben a tartalomjegyzéket.* Az első sorban levő 0 karakternek nincs

* Valójában a képernyőn megjelenő kiírás a lemez tényleges tartalomjegyzékének csak a fejléce. L. 3.4.3. (A ford. megjegyzése.)

különösebb jelentősége. Mellette látható a lemez neve és ID-je, amiről a formálás ismertetése során már szó esett. A 2A karakterek szimbolizálják a lemezformátumot. Ha ez a formátum nem 2A, akkor ezt a lemezt nem ezen az egységtípuson formálták, így ezzel az egységgel nem is használható.
A következő sorokban a lemezen levő file-ok adatai láthatók. A file blokkhosszúsága a sor elején, típusa pedig a sor végén olvasható. Ezen a lemezen három különböző file-típus található:

PRG – ezek a programfile-ok, vagyis a BASIC vagy gépi nyelven írt programok
SEQ – ez a soros file-ok jelölése, amelyekről a későbbiekben bőven lesz szó
REL – ez az adattárolásnak egy másik formája, amelynek részletes ismertetésére szintén a továbbiakban kerül sor

A file-ok hossza blokkban van megadva, a blokkok mindegyike 256 byte-ot foglal magában. Így könnyen meghatározható egy program nagysága. Az egyes blokkok 256 byte-jából mindössze 2 byte-ot kell leszámítani, nevezetesen azokat, amelyek az egyes blokkok láncolásához szükségesek.
Az utolsó sorban a lemez szabad blokkjainak száma látható. Ha a file-ok hosszát összeadjuk, majd ehhez hozzáadjuk a szabad blokkokat, megkapjuk a lemezen lefoglalható blokkok számát (664).
Ha van nyomtatónk, a tartalomjegyzéket programlista formájában ki tudjuk nyomtatni. Ehhez az alábbi utasítássorozatot használhatjuk:

OPEN 1,4	megnyitja a nyomtatót
CMD 1	a vezérlést átadja a nyomtatóra*
LIST	kinyomtatja a tartalomjegyzéket
PRINT#1	egy RETURN érkezik a nyomtatóhoz
CLOSE 1	ismét lezárja a nyomtatót

A nyomtatásnak értelemszerűen előfeltétele, hogy a tartalomjegyzéket LOAD "$",8 paranccsal előzetesen tényleg betöltsük a tárba. Ha a tárban BASIC program van, az is ezzel a rutinnal nyomtatható ki.
Az 1.3.10. alfejezetben bemutatjuk, hogyan lehet a tartalomjegyzéknek csak egy részét kiíratni, ha csak bizonyos programokra vagyunk kíváncsiak.

1.3.5. A SCRATCH – file-ok törlése

Azokat a file-okat, amelyekre a további munkánk során nincs szükségünk, törölhetjük a lemezről. Erre szolgál a SCRATCH utasítás. Mielőtt ezt az utasítást alkalmaznánk, okvetlenül győződjünk meg arról, hogy a SCRATCH utasításban megadott név a törlendő file nevével megegyezik-e. Ha tévedésből törlünk egy file-t, ezáltal több órás, sőt több napos munkánk is veszendőbe mehet.
A törlő utasítás formátuma:

PRINT#lfn, "SCRATCH: FILE-NÉV1, FILE-NÉV2,..."

* Elsődleges output egység a képernyő, alapértelmezésben minden kiíró művelet (PRINT, LIST) erre vonatkozik. A CMD utasítás ezt az állapotot módosítja. (A ford. megjegyzése.)

Több file is törölhető egyetlen utasítással. Az egy utasítással törölhető file-ok számának csak az szab határt, hogy a PRINT utasítás után 40 karakternél többet nem írhatunk.
A következő példa a TESZT nevű file-t törli a lemezről:

```
OPEN 1,8,15,"S:TESZT"
CLOSE 1
```

Ha a 15-ös csatornát előzőleg már megnyitottuk, elegendő az alábbi PRINT utasítás:

```
PRINT#1, "S:TESZT"
```

Az egész lemez tartalmát is törölhetjük. Ehhez az 1.3.10. alfejezetben leírt JOKER (* – csillag karakter) használata szükséges:

```
PRINT#1, "S:*"
```

Végzetes lehet a fenti utasítás hatása, ha előzőleg nem győződtünk meg arról, hogy nincs számunkra egyetlen használható file sem a lemezen.
A törlő utasítás után a hibacsatornából érkező üzenet:

```
01 FILES SCRATCHED nn 00
```

ahol az "nn" adja meg a törölt file-ok számát. Ez az üzenet az 1.3.3. alfejezetben lévő rutinnal olvasható ki.

1.3.6. A RENAME – file-ok átnevezése

Ahhoz, hogy egy file másik nevet kaphasson, a file-bejegyzésben meg kell változtatni a file nevét. Erre szolgál a RENAME utasítás, amelynek formátuma a következő:

```
RENAME: új név=régi név.
```

Ha például a file nevét TESZT-ről TESZT.01-re kell módosítani, úgy az alábbi utasításokat használjuk:

```
OPEN 1,8,15,"R:TESZT.01=TESZT"
CLOSE 1
```

vagy

```
OPEN 1,8,15
PRINT#1, "R:TESZT.01=TESZT"
CLOSE 1
```

Egy már megnyitott, de még le nem zárt file-nak nem adható új név!

1.3.7. A COPY – file-ok másolása

Ezzel az utasítással egy file egy lemezen belül másolható. Több soros (szekvenciális) file-ból képezhetünk így egy file-t. Ha pl. a háztartásunkban a kiadásokat havi bontásban soros file-okban tároljuk és ezeknek az KIAD.01, KIAD.02 stb. nevet adjuk, egyetlen utasítással kialakíthatjuk az év első negyedében esedékes (például KIAD.N1) kiadásaink file-ját. Mivel az utasítás formátuma:

COPY: új file=régi file1, régi file2,......,

az említett file-ok összevonását az alábbi utasításokkal végezhetjük el:

```
OPEN 1,8,15, "C:KIAD.Ø1=KIAD.Ø1,KIAD.Ø2,KIAD.Ø3"
CLOSE 1
```

Ezt az összefűzési módszert a programoknál nem alkalmazhatjuk. Egy lemezen belül csak egy programot tudunk átmásolni, és azt is csak akkor, ha a másolással kapott új program neve még nem szerepel a tartalomjegyzékben.
A COPY utasítást ritkán használjuk. Ennek oka, hogy a file-nak ugyanarra a lemezre történő másolása lényegében értelmetlen. Ennek az utasításnak az egyetlen ésszerű felhasználása az az eset, amikor több soros- vagy user file-t* kapcsolunk össze adathalmazzá. Ezzel szemben gyakran szükség van arra, hogy egy file-t egy másik lemezre másoljunk át. Ez elengedhetetlenül fontos az adatok biztonságos tárolásához. Ha két lemezegységünk van, a COPY/ALL program segítségével a file-ok átmásolhatók az egyik egységről a másikra, de csak akkor, ha az egyik egység száma 9 (a COPY/ALL program rajta van a TEST/DEMO lemezen).
Van persze másolási lehetőség azok számára is, akik csak egy egységgel rendelkeznek. Ez a felhasználói kör a 4.1. fejezetben ismertetett kiszolgálóprogramokkal az egyes file-okat, sőt az egész lemezt is átmásolhatja.

1.3.8. Az INITIALIZE – a lemez inicializálása

A DOS a lemez minden blokkjáról nyilvántartja, hogy az foglalt vagy sem. Ezt a nyilvántartást röviden BAM-nak nevezzük (Block Availability Map). Ha egy lemezzel éppen dolgozunk, a DOS a lemezműveleteket mindig az egység RAM területén lévő BAM alapján végzi el, ill. ezen hajtja végre a szükséges módosításokat. Ha munka közben lemezt cserélünk, ezt a DOS a lemez azonosítója (ID) alapján „észreveszi". Ha azonban az új lemez ID-je azonos a korábbi lemez ID-jével, a DOS nem figyel fel a lemezcserére. Ugyanakkor az első lemeznek a még a tárban lévő BAM-ja már nem azonos a most behelyezett lemez BAM-jával. Így azok az utasítások, amelyek nem a végrehajtás előtt olvassák be a BAM-ot a tárba (például az összes közvetlen elérésű utasítás), a lemezegység tárában lévő BAM-ot használják (hibásan!) a blokk foglaltsági állapotának adminisztrálására.
Megszívlelendő tehát az a javaslat, hogy formáláskor sohase használjunk azonos ID-ket.
Az INITIALIZE utasítás betölti az éppen az egységben levő lemez BAM-ját a lemeztárba. Az utasítás formátuma a következő:

* L. az 1.4.2. és az 1.4.6. alfejezeteket

PRINT#lfn, "INITIALIZE"

vagy rövidített formában:

PRINT#lfn, "I"

Példa:

```
OPEN 1,8,15,"I"
CLOSE 1
```

A fent leírt súlyos hibák elkerülése érdekében célszerű minden lemezcsere után kiadni az inicializálási utasítást, és ez vonatkozik arra az esetre is, amikor egy program futása közben kell lemezt cserélni.

1.3.9. A VALIDATE – a lemez érvényesítése

A VALIDATE utasítás a lemez összes olyan foglaltnak minősített blokkját, amely ugyanakkor egyetlen „élő" file-hoz sem tartozik, ismét felszabadítja. Ha például megnyitunk a lemezen egy file-t, az adatokat beleírjuk, de CLOSE utasítással nem zárjuk le, a file VALIDATE (érvényesítés) esetén ismét törlődik. Hasonlóan, látszólag foglalt blokkokat állítunk elő, ha közvetlen lemezművelettel (Block-Write) a blokkokat foglaltnak minősítjük. Ezek a blokkok valójában egyetlen file-hoz sincsenek hozzárendelve és így az érvényesítés során felszabadíthatóak.
A VALIDATE utasításnak van még egy fontos funkciója. Ha a lemezről SCRATCH utasítással törlünk egy file-t, a DOS valójában nem hajtja végre a törlést, csak a file-típust változtatja meg „törölt file-típusra", tehát a tartalomjegyzékben a file-ra vonatkozó bejegyzés – a file neve – a továbbiakban nem jelenik meg. Az ilyen módon törölt file-t egy gyakorlott programozó a Disk-Monitor programmal újra helyreállíthatja, hiszen a file tartalma sértetlenül ott van a lemezen.
Érvényesítés során ezek a látszólagosan törölt file-ok ténylegesen törlődnek a lemezről.
Az utasítás formátuma:

PRINT#lfn, "VALIDATE"

vagy rövidített formában:

PRINT#lfn, "V"

Egy példa:

```
OPEN 1,8,15, "V"
CLOSE 1
```

A DOS fentiekben leírt lemezkezelő sajátosságai miatt előfordulhat, hogy egy lemezen látszólag több, mint 664 blokk van. Természetesen ez nem igaz, a lemez érvényesítése után értesülhetünk a tényleges helyzetről.

Az ellenkező eset is előfordulhat. Egy file tárolása közben a DISK FULL üzenetet kapjuk, pedig a tartalomjegyzék alapján kiszámolt szabad blokkokra elférne a file. Az érvényesítés ekkor is segít.

1.3.10. ?* – a "Joker"

Két joker karaktert használhatunk, a csillagot (*) és a kérdőjelet (?). A csillag a file-név egy meghatározott helyén azt jelenti, hogy a lemezen arra a sorrendben első programra vagyunk kíváncsiak, amelynek a neve a csillag előtti karakterekkel kezdődik. Egy példa:

```
LOAD "TESZT*",8
```

Ez az utasítás betölti az első programot, amelynek az első öt betűje "TESZT". A

```
LOAD"*",8
```

utasítás betölti a lemez első programját, mivel a csillag előtt nem adtunk meg egyéb karaktert. A SCRATCH utasításban a csillag más funkciót lát el. Itt nem az első file törlésére kerül sor, hanem az összes olyan file törlődik, melynek neve a csillag előtti karakterekkel kezdődik. Az

```
OPEN 1,8,15, "S:TESZT*"
CLOSE 1
```

utasítás például a "TESZT" betűkkel kezdődő összes file-t törli. Ezt mindenkor tartsuk szem előtt! Az utóbbi lehetőséget a tartalomjegyzék betöltésénél is használhatjuk. A

```
LOAD "$A*",8
```

utasítás a tartalomjegyzéknek csak azt a részét tölti be, amelyben a file-ok neve az "A" betűvel kezdődik.
A csillag használatával a file-típusok is kiválogathatók, ha a csillag után egy egyenlőségjelet teszünk s ezután az első betűvel a kívánt file-típusra utalunk. Egy ilyen válogatás:

*=S	csak a soros file-okat válogatja ki
*=P	a programfile-okat válogatja ki
*=R	a relatív file-okat válogatja ki
*=U	a user file-okat válogatja ki

Írjuk be például a

```
LOAD "$*=P",8
```

utasítást. Betöltés után a LIST parancsra csak a programfile-ok jelennek meg a tartalomjegyzékben.
A SCRATCH utasítással például a lemezen lévő összes soros file az alábbi utasítással is törölhető:

```
OPEN 1,8,15, "S:*=S"
CLOSE 1
```

A csillag előtt természetesen egy jelsorozatot is megadhatunk, s így csak azokat a soros file-okat töröljük, amelyeknek a neve ezzel a jelsorozattal kezdődik.
A kérdőjellel a betűk a file-névben tetszőleges helyen „nem érdekes"-nek minősíthetők.
A kérdőjel funkciójának megértéséhez nézzük a következő példákat:

A?????	– keressük azokat az 5 helyértékű file-neveket, amelyeknek az első betűje "A",
????TEST	– keressük azokat a 8 helyértékű file-neveket, amelyeknek az utolsó négy betűje "TEST".

A csillag és a kérdőjel kombinálása is megengedett. Ebben az esetben ne feledkezzünk meg arról, hogy a csillag után sem betű, sem pedig kérdőjel nem következhet, miután ezek a kombinációk értelmetlenek. Nézzük meg a csillag és a kérdőjel társításának két példáját:

????.*	– keressük az összes olyan file-nevet, amelynél a pont előtt 4 tetszőleges karakter áll.
TEST.??*	– keressük az összes olyan, legalább 7 helyértékű file-nevet, amelyeknél az első 5 karakter TEST.-ot tartalmaz.
TEST–??Ø1*=S	– keressük az összes olyan, legalább 9 helyértékű soros file-t, amelynek a nevében az első 5 helyértékben TEST-, a 8. és 9. helyértéken pedig Ø1 áll.

1.4. A soros adattárolás

A lemezegység nemcsak programtárolásra használható. Előbb-utóbb minden kezdő haladóvá válik és nagy adatmennyiségek kezeléséhez akar saját programokat írni. Bár a soros adattárolás nem a leggyorsabb, de mégis a legegyszerűbb adatkezelési eljárás, ami főleg a „közép-haladók" számára fontos. Ez az adatszervezés hasonló a mágnesszalagos soros adatfeldolgozáshoz, melynek logikai felépítése lehet pl. a következő:

1. a program betöltése
2. a teljes adatállomány betöltése a tárba
3. adatok kezelése a tárban (változtatás, törlés, hozzátoldás)
4. a felújított állomány tárolása háttértárolón (szalagon, lemezen)
5. kilépés a programból

Ezt a szervezési módot használva a maximális adatmennyiség a gép tárkapacitásának függvénye, miután egy soros file-ban levő állomány egyes adatai közvetlenül nem módosíthatóak, vagy törölhetőek. Ehhez az egész file beolvasása, megváltoztatása és újratárolása szükséges.* Lemezes tárolás esetén a file betöltése és tárolása gyorsabb a szalagos tárolásnál. Ez a lemezes adattárolás egyik előnye.
Másik nagy előnye, hogy a lemezen tárolt soros adatállomány bővítéséhez nem szükséges az egész file beolvasása. A lemezes soros file-t ugyanis megnyithatjuk bővítésre is (APPEND-re), ami kazettás egységen nem lehetséges.

* A módosítás megoldható úgy is, hogy folyamatos feldolgozással átmásoljuk a megváltoztatott adatokat egy másik file-ba, de ez nagy adathalmaznál rendkívül lassú eljárás. (A ford. megjegyzése.)

A felhasználók számára fontos tény, hogy az eddig kazettás szalagon soros állományokkal dolgozó programok egyszerűen átalakíthatóak lemezen dolgozó programokká. Ehhez csak a megfelelő OPEN utasítások megváltoztatása szükséges.

1.4.1. A soros tárolás alapelve

A soros file több rekordból tevődik össze, amelyek mezőkre tagolódnak. Egy címfile példáján ezt könnyen megérthetjük, amelyben egyes címek képezik a file rekordjait. Egy címtétel több mezőből áll (vezetéknév, keresztnév stb.). A file szerkezete az alábbiak szerint ábrázolható:

1. MEZŐ:	2. MEZŐ:	3. MEZŐ:	1. MEZŐ:	2. MEZŐ:	3. MEZŐ:
1. REKORD			2. REKORD		
FILE					

Egy file rekordjai ugyanúgy, mint a rekordokban a mezők, egymást követve, vagyis sorban helyezkednek el. A mezők, s így a rekordok is különböző hosszúságúak lehetnek. Pl. az 1. rekord 1. mezőjének hossza meghaladhatja a 2. rekord 1. mezőjének hosszát. Ez a lehetőség abból adódik, hogy a mezőket egy RETURN karakter választja el egymástól, amit a PRINT utasítás állít elő és az INPUT utasítás ismer fel. Adatkezeléskor általában minden mezőhöz egy változót rendelünk, ezeket a PRINT/INPUT utasításokkal írjuk, olvassuk. Egyetlen utasítással azonban egy egész rekordot is olvashatunk vagy írhatunk. Ennek feltétele az, hogy az összes rekord azonos hosszúságú legyen, miután a programon belül ezeket a rekordokat speciális utasításokkal mezőkké kell felbontani. Ez csak akkor lehetséges, ha a rekordon belül minden mező kezdőpozícióját pontosan ismerjük.
Az elmondottakat egy telefon-file bemutatásával szemléltetjük. Tegyük fel, hogy minden előfizetőről három adatot tartunk nyilván:

1. MEZŐ: Vezetéknév
2. MEZŐ: Keresztnév
3. MEZŐ: Telefonszám

A file a következőképpen nézhet ki (a RETURN helyett egy + jelet írunk):

Karakterek:	11111111112222222222333333333344444444445 1234567890123456789012345678901234567890
Adat:	SZABO+ANNA+236+BIRO+PÁL+1213+KOVÁCSI+ÉVA+65432+...

Mint látható, a különböző hosszúságú mezőket a rekordon belül egy RETURN választja el egymástól. Ez a RETURN egy PRINT utasítással történő átvitelkor az adatok mögé kerül, hacsak az említett PRINT utasítást nem pontosvessző (;) követi, amely a RETURN-t elnyomja. Az első INPUT utasítással a rekord első karakterei bekerülnek egy változóba, egészen a RETURN-ig. A következő INPUT beolvassa a következő mezőt a következő változóba ismét a RETURN-ig stb. Ezután az ízelítő után az 1.4.2. alfejezetben minden olyan programozástechnikai eszközt részletesen elemzünk, ami a soros adatállományok kezeléséhez szükséges.

1.4.2. A soros file megnyitása

Minden file-t létrehozás előtt meg kell nyitni. A soros file-t megnyithatjuk írásra, olvasásra és bővítésre. Ha írásra nyitjuk meg a DOS az alábbi műveleteket végzi el:

1. ellenőrzi, hogy a lemezen van-e már ilyen nevű file, ha van, a rendszer a FILE EXISTS hibaüzenetet küldi,
2. a hiba kiküszöbölése után a megfelelő *file-bejegyzést* beírja a tartalomjegyzékbe. Listázáskor, ha egy file-t megnyitottunk, de nem zártunk le, a tartalomjegyzékben a file-típus előtt egy csillag jelenik meg,
3. keres egy szabad blokkot, amelyben az első adatok tárolhatók. A címeket (sáv és szektor) a file-bejegyzésben tárolja,
4. a file által lefoglalt blokkok számát 0-ra állítja, mivel a file még egyetlen blokkot sem tartalmaz.

A létrehozott, adatokkal feltöltött soros állományt módosíthatjuk vagy bővíthetjük. Az OPEN utasításban határozzuk meg, hogy a file-t milyen célra akarjuk megnyitni. Az OPEN utasítás formátuma a következő:

```
OPEN lfn,8,sa, "file-név, file-típus, mód"
```

Ha azt akarjuk, hogy egy file-ra vonatkozó PRINT utasítás csak egy RETURN-t írjon a file-ba, a logikai file-számot 1–127 között kell megválasztani. Ha ugyanis a logikai file-szám nagyobb 127-nél (128–255), a PRINT utasítás minden egyes RETURN után még egy „soremelés" (LINE-FEED) karaktert is továbbít. Ez például olyan nyomtatóknál szükséges, amelyeknél RETURN után nincs automatikus soremelés.
Az "sa" az ún. másodlagos cím (secondary address); ez jelöli a lemezegységnek azt a csatornáját, amelyen keresztül az adatok továbbítódnak. A 0 és az 1 másodlagos címet az operációsrendszer a programok tárolásához és betöltéséhez tartja fenn. A 15-ös másodlagos cím az utasítás- és a hibacsatornára vonatkozik. Ha egyidejűleg több file-t nyitunk meg, a logikai file-szám mellett a másodlagos címeknek is feltétlenül eltérőeknek kell lenniük, mert *egy* file-ra nézve mindig csak *egy* adatcsatorna lehet illetékes. Ha azonban egy file-t olyan másodlagos címmel nyitunk meg, amellyel előzőleg már megnyitottunk egy másik file-t, úgy az első file lezárul. Gyakran szokták figyelmen kívül hagyni azt a feltételt, hogy minden egységre maximum három csatorna nyitható meg. A *relatív file*-ok kezeléséhez a DOS-nak azonban egyidejűleg két csatornára van szüksége. Ennek megfelelően tehát az alábbi kombinációkról lehet szó:

- 1 relatív és 1 soros file,
- 3 soros file.

A file-név megadásakor ne feledkezzünk meg annak ellenőrzéséről, hogy ilyen nevű file létezik-e már a lemezen. Ha már létező file-t akarunk írásra megnyitni, a SAVE utasításhoz hasonlóan itt is az @ jelet kell a file-név elé beírni, amit egy kettőspont követ. Például:

```
OPEN 1,8,2,"@:CÍMEK,S,W"
```

A file megnyitásakor mindig meg kell adni a file-típust. Az OPEN utasításban a file-típusok rövidítése a következő:

S – soros file,
U – user file,
P – programfile
R – relatív file.

A user file-ok olyan soros file-ok, amelyek a tartalomjegyzékben USR file-típussal szerepelnek, és tulajdonképpen nem is file-ok. Ez a file-típus jól alkalmazható azokban az esetekben, amikor lemezen kell tárolni az egyébként képernyőn megjelenítendő, vagy nyomtatásra kerülő szövegeket (BASIC lista vagy tartalomjegyzék). Ennek az eljárásnak a leírása az 1.4.6. alfejezetben olvasható.
Az utolsó paraméter (modus) határozza meg az adott csatorna használatának mikéntjét. Az alábbi négy lehetőség áll rendelkezésre:

W – egy file írása (WRITE – 1.4.3. alfejezet)
R – egy file olvasása (READ – 1.4.4. alfejezet)
A – egy soros file bővítése (APPEND – 1.4.4. alfejezet)
M – egy, még nem lezárt file olvasása (1.4.5. alfejezet)

Nyissuk meg írásra a "SEQU.TEST" nevű soros file-t:

```
OPEN 1,8,2,"SEQU.TEST,S,W"
```

Ha ezután a LOAD "$",8-cal a tartalomjegyzéket betöltjük és kilistázzuk, látunk a file-típus előtt egy csillagot, ami arra utal, hogy a file létezik, de nem használható, mert elfelejtettük lezárni:

```
Ø SEQU.TEST *SEQ
```

A továbbiakban sajnos már nem is tudjuk lezárni. Jegyezzük meg tehát, hogy egy file megnyitása után, a tartalomjegyzék betöltése előtt elengedhetetlen a file lezárása!
Ha egy file-t megnyitottunk, megnyithatjuk a parancs/hibacsatornát is. A hibacsatorna lezárása után azonban minden nyitott file automatikusan lezáródik. Erről nem szabad megfeledkezni!

OPEN 1,8,2,"SEQU.TEST,S,R"	– egy soros file megnyitása olvasásra
OPEN 2,8,3,"SEQU.TEST,U,W"	– egy userfile megnyitása írásra
OPEN 3,8,4,"TEST,P,R"	– egy programfile megnyitása olvasásra
OPEN 4,8,5,"SEQU.TEST,S,A"	– egy soros file megnyitása bővítésre
OPEN 5,8,6,"UGYFEL.1983,S,M"	– a felhasználói file nem szabályosan zárult le, így újra kell olvasni.

1.4.3. A lemezegység és az alapgép közötti adatátvitel

Egy írásra megnyitott file-ba PRINT utasítással írhatunk adatokat. A PRINT utasítás végrehajtásával a DOS egy RETURN karaktert is továbbít, amely az adatok elválasztásához szükséges. Az alábbiakban egy file megnyitására, írására és lezárására mutatunk példát. Az utasítások közvetlenül (sorszám nélkül) is használhatók, így egymás után begépelhetjük, majd egyenként végrehajthatjuk őket. Nyissuk meg a TEST.1 nevű file-t írásra:

OPEN 1,8,2,"TEST.1,S,W"

A lemezegységen kigyullad a vörös színű LED. Ez azt jelenti, hogy egy file megnyitott állapotban van, vagyis írhatunk bele adatokat. Példaként az előző címfile egy rekordját (amely négy mezőből áll) írjuk fel lemezre az alábbi PRINT# parancsokkal:

```
PRINT#1,"KOVACSI"
PRINT#1,"ANNA"
PRINT#1,"POZSONYI UTCA 7"
PRINT#1,"5900 BEKESCSABA"
```

A végrehajtás után a file-t a CLOSE 1 paranccsal lezárhatjuk. A vörös színű LED ezzel egyidejűleg kialszik. Ha vissza akarjuk olvasni a lemezről ezeket az adatokat, a file-t olvasásra kell megnyitni (R). Mivel az INPUT utasítás parancsként nem használható, egy kis programot kell írnunk:

P. 3.

```
10 open1,8,2,"teszt.1,s,r"
20 input#1,vn$
30 input#1,nn$
40 input#1,st$
50 input#1,or$
60 close1
70 print"vezeteknev:       ";vn$
80 print"keresztnev:       ";nn$
90 print"lakascim,utca:    ";st$
100 print"            ,hszam: ";or$
```

A program magyarázata egyszerű:

10-es sor	a TEST.1 file-t megnyitjuk olvasásra
20–50. sor	az adatokat visszaolvassuk a felírás sorrendjében
60-as sor	a file-t lezárjuk
70–100. sor	az adatokat megfelelő kísérőszöveggel együtt kiírjuk a képernyőre.

Ha ezt az utasítássorozatot begépeljük és RUN-nal indítunk, a képernyőn megjelennek a file-ba előzetesen beírt adatok:

vezetéknév:	KOVACSI
keresztnév:	ANNA
utca:	POZSONYI UTCA 7
irányítószám/helység:	5900 BEKESCSABA

Az adatok beolvasása négy INPUT utasítást igényel, miután egy rekord négy mezőből áll. Ha például egy olyan file-t kell tárolnunk, amelynek rekordjai 20 mezőből tevődnek össze, a beolvasáshoz 20 INPUT utasításra van szükség. Ciklusutasítás használatával a feladat lényegesen leegyszerűsíthető:

P. 4.

```
10 open1,8,2,"teszt.1,s,r"
20 fori=1 to 4
30 input#1,d$(i)
40 next
50 close1
60 print"vezeteknev:        ";d$(1)
70 print"keresztnev:        ";d$(2)
80 print"lakascim,utca:     ";d$(3)
90 print"           ,hszam: ";d$(4)
```

A négy füzér (string) helyett egy négy elemű indexes változót használtunk. Ne feledkezzünk meg arról, hogy a BASIC 2.0-nál az index legfeljebb 10 lehet, ha a tömböt DIM utasítással előzetesen nem deklaráltuk nagyobb indexhatárral. Ha a példánkban egy 20 mezős rekordot akarunk beolvasni, a tömböt először deklaránunk kell a DIM D$(20) utasítással. A beolvasás és kiírás rövidítésére még egy további lehetőség is rendelkezésre áll: a billentyűzetről történő adatbevitelnél az INPUT utasítással több, egymástól vesszővel elválasztott változót beolvashatunk. Például:

INPUT VN$,NN$,TE

Ennél az utasításnál a három változót például az alábbiak szerint kell beírnunk:

MOLNAR,LAJOS,7465

A beolvasott adatokat ekkor a

PRINT VN$,NN$,TE

ismét megjeleníti a képernyőn.
A képernyőre illetve a soros file-ra vonatkozó PRINT utasítások között az egyetlen különbség, hogy az utóbbinál a füzéreket idézőjel közé zárt vesszőkkel kell elválasztani egymástól. Ha például az említett változókat egy file-ba kell beírni, úgy ehhez a PRINT utasítást az alábbiak szerint kell módosítani:

PRINT#1,VN$","NN$","TE

A numerikus változókat csak vesszővel kell elválasztani a többi változótól. Ebben az esetben az adatok beolvasásához az

INPUT#1,VN$,NN$,TE

utasítást használjuk. Miután azonban az INPUT utasítással bevihető karakterek száma maximálisan 88, ez korlátot szab az ilyen típusú adatbevitel használatának. Ha egy rekordon belüli mező hossza meghaladja a 88 karaktert, a beolvasáshoz másik utasítást kell alkalmaznunk. Ez a GET utasítás, amely az egyes karaktereket külön-külön olvassa be. Tegyük fel, hogy egy olyan rekordot akarunk olvasni, amely egyetlen 100 karakter hosszúságú mezőből áll. Ez a rekord az alábbi rutinnal olvasható be a lemezről egy füzérbe:

```
10 OPEN 1,8,.........
20 D$=""
30 FOR I=1 TO 100
40 GET#1,X$
50 D$=D$+X$
60 NEXT I
70 GET#1,X$
80 CLOSE 1
```

A fenti utasítássorozat lefutását követően a D$ füzér egy 100 karakteres rekordot fog tartalmazni. Minden soros file megnyitása után a DOS beállít egy *mutatót* (pointert), amely mindig az eddig olvasott adatok utáni karakterre mutat. Tegyük fel, hogy a 100 karakteres rekordot lezáró pontosvessző nélkül írtuk be PRINT-tel a file-ba, így a rekordot egy RETURN zárta le. A 100. karakter olvasása után a mutató ezen a RETURN karakteren áll. A következő GET utasítás a 70. sorban tehát azért szükséges, hogy ezt a RETURN-t is beolvassuk. Így a következő rekord olvasására szolgáló első GET utasítás nem a RETURN-t, hanem valóban a következő rekord első karakterét éri el.
A fenti példában feltételeztük, hogy a file rekordjainak hossza állandó, 100 karakter. Általában azonban a soros file-ban szereplő rekordok hossza nem konstans. Ha a rekord maximális hosszúsága túllépi a 88 karakteres input határt, az adatbevitelt GET ciklussal kell megszervezni, amely az elválasztó RETURN-t rekordvégként észleli.
Egy ilyen rutin a következőképpen néz ki:

```
10 OPEN 1,8,.........
20 S$=""
30 GET#1,X$
40 IF X$=CHR$(13) THEN 80
50 S$=S$+X$
60 IF ST<>64 THEN 30
70 CLOSE 1:END
80 PRINT S$
90 GOTO 20
```

A program egy változó hosszúságú rekordokból álló file olvasását és a képernyőn történő megjelenítését végzi el.
Ha el akarjuk kerülni a 88 karakternél hosszabb rekordokkal kapcsolatos problémáinkat, célszerű felosztani a rekordot több részre, majd beolvasás után a részeket összekapcsolni.

1.4.4. A soros file bővítése

Ha a soros file bővítését úgy kellene elvégeznünk, hogy először beolvassuk a teljes adatállományt a tárba, ott kibővítjük, majd a bővített állományt visszaírjuk a lemezre, ez bizonyára rendkívül sok időt venne igénybe. A DOS szerencsére kényelmes lehetőséggel szolgál a soros file bővítésére anélkül, hogy a file-t előzetesen be kellene olvasni a tárba. A file-t ekkor APPEND-re (bővítésre) kell megnyitnunk, azaz az OPEN utasításban az A üzemmódot kell választanunk.
Gépeljük be a következő utasításokat:

```
OPEN 1,8,2,"TEST.2,S,W"
PRINT#1,"1.REKORD"
CLOSE 1
```

Ezzel egy 1 rekordot tartalmazó soros file-t állítottunk elő. Bővítsük most a file-t további két rekorddal:

```
OPEN 1,8,2,"TEST.2,S,A"
PRINT#1,"2.REKORD"
PRINT#1,"3.REKORD"
CLOSE 1
```

A TEST.2 file most már három rekordot tartalmaz. Ezt az alábbi programmal ellenőrizhetjük:

P. 5.

```
100 open 1,8,2,"test.2,s,r"
110 fori=1 to 3
120 input#1,ds$
130 print ds$
140 close 1
```

A fenti program beolvassa a rekordokat a file-ból és megjeleníti őket a képernyőn.
Az A üzemmód soros adatszervezés esetén gyors adatbővítést (file-bővítést) tesz lehetővé.

1.4.5. A soros file lezárása

A CLOSE utasítással lezárhatjuk a már megnyitott file-okat. Az utasítás formátuma a következő:

```
CLOSE lfn
```

Az lfn paraméter a megfelelő OPEN utasításnál megadott logikai file-szám. Amennyiben több file-t kell lezárnunk, minden egyes file-hoz egy külön CLOSE utasítást kell használni. Az utolsó file lezárásával a lemezegységen kialszik a vörös színű LED. Adatátvitelnél az adatok egy csatornán keresztül érkeznek a lemezegységhez. Ez a csatorna egy tárolón belüli tár *(puffer)*, ahol a géptől érkező adatok közbenső tárolása történik. A pufferben lévő adatok csak akkor továbbítódnak a lemezre, ha ez a puffer már megtelt.
A file lezárásakor a lemezre íródnak azok az adatok, amelyek még a pufferben voltak. Egy nem lezárt file nem teljes, mivel a puffer nem ürült ki, így a DOS a file-t szabálytalannak minősíti és nem engedi READ-re megnyitni; ilyen kísérletnél a WRITE FILE OPEN üzenetet küldi.
Persze rendkívül bosszantó lenne, ha egy adathalmaz véglegesen elveszne a felhasználó számára, ezért a DOS tartalmaz egy lehetőséget az ilyen jellegű hibák áthidalására. A szabálytalannak minősített file-ok olvasására szolgál az M üzemmód. Célszerűbb megoldás természetesen a file-ok szabályos lezárása.
A következő program egy szabálytalan file tartalmát „átmenti" egy normál file-ba:

P. 6.

```
100 input"eredeti file";u$
110 input"cel file    ";z$
120 open 1,8,2,u$+",s,m"
130 open 2,8,3,z$+",s,w"
140 input#1,x$
150 print#2,x$
160 if st<>64 then 140
170 close 1:close 2
180 open 1,8,15,"s:"+u$
190 close 1
```

A program végén a rossz file törlődik.

1.4.6. Elsődleges output átirányítása lemezre

Minden parancsként (sorszám nélkül) vagy utasításként kiadott LIST vagy PRINT alapértelmezésben a képernyőre vonatkozik, azaz az output a képernyőn jelenik meg.
Ezt az állapotot megváltoztathatjuk CMD utasítással, melynek formátuma:

CMD lfn,

ahol az lfn egy logikai file-számot jelöl. Az utasítás használata előtt az adott logikai file-t meg kell nyitni.
Elsődleges output egységként használhatjuk a lemezegységet is azzal a megszorítással, hogy a lemezen megnyitott file-nak ún. USR (user) típusú soros file-nak kell lennie azért, hogy a DOS meg tudja különböztetni a normális soros file-októl.
A következő parancsokkal a tárban lévő programot *szöveg-file*-ként tároljuk a TEST.LIST nevű soros USR típusú lemezes file-ban:

```
OPEN 1,8,2,"TEST.LIST,U,W"
CMD 1
LIST
CLOSE 1
```

A CLOSE 1 utasítás végrehajtásakor megszűnik a CMD utasítás hatása, azaz az elsődleges output ismét a képernyő lesz.
Mire használhatjuk ezt a lehetőséget?
Tegyük fel, hogy egy szövegfeldolgozó programmal olyan dokumentációt szeretnénk készíteni, amelyben mintaprogramok is szerepelnek. Eddigi tudásunk alapján a feladatot úgy oldanánk meg, hogy a szövegrészeket lemezen tároljuk, a mintaprogramokat LIST paranccsal kinyomtatjuk és végül a megfelelő szövegrészekbe a programlistákat kézzel beszerkesztjük. Sokkal kényelmesebb megoldás azonban, ha a programokat is szövegként tároljuk, hiszen így a szövegszerkesztő programmal nyomtatásra kész dokumentációt állíthatunk elő.
Arra is van magyarázat, hogy miért van szükségünk az USR file-típusra, miért nem kezelhetjük a programfile-t szövegként. A programok tárolása ugyanis nem karakteresen történik, hanem utasításszimbólumokkal (ún. *token*ekkel), amelyeket a szövegszerkesztő program nem ismer.
Az alábbi rutinnal visszaolvashatjuk és kiírathatjuk a lemezen tárolt szöveg-file-t:

P. 7.

```
10 open 1,8,2,"teszt.list,u,r"
20 get#1,x$
30 print x$;
40 if st<>64 then 20
50 close 1
```

A rutin lényegében egy ciklus, amely a file összes karakterét (byte-ját) végigolvassa és kiírja a képernyőre. A file vége az ST (status) *rendszerváltozó* alapján ismerhető fel, amelynek értéke az utolsó karakter beolvasásakor 64, egyébként 0. A soros file nyomtatásához az alábbi utasítássorozat szükséges:

P. 8.

```
10 open 1,8,2,"teszt.list,u,r"
20 open 2,4
30 get#1,x$
40 print#2,x$;
50 if st<>64 then 30
60 close 1
```

A 20-as sorban megnyitottuk a 4-es egységszámú nyomtatót (hozzárendeltük a 2-es logikai file-hoz).

1.4.7. Soros file táblázatos kezelése a tárban

Ha egy adathalmazt soros file-ként tároltunk és olyan jellegű feldolgozást akarunk elvégezni, amelyhez az adathalmaz összes elemére szükségünk van – mivel a file-t csak sorosan tudjuk végigolvasni – feldolgozás előtt az egészet be kell olvasnunk a tárba.* Mielőtt a beolvasás megkezdődne, DIM utasítással helyet kell foglalni (deklarálni kell) a tárban az indexes változó számára. Rekord szerkezetű adathalmaz kezelésére alkalmas a kétindexes változó (táblázat), melynek első indexe a rekord–, a második pedig a rekordon belüli mező sorszáma. Tekintsünk egy ilyen táblázatot:

	1. mező	2. mező	3. mező
1. rekord	D$(1,1)	D$(1,2)	D$(1,3)
2. rekord	D$(2,1)	D$(2,2)	D$(2,3)
3. rekord	D$(3,1)	D$(3,2)	D$(3,3)
4. rekord	D$(4,1)	D$(4,2)	D$(4,3)
5. rekord	D$(5,1)	D$(5,2)	D$(5,3)
6. rekord	D$(6,1)	D$(6,2)	D$(6,3)

* Ha az adatok feldolgozásának sorrendje azonos a tárolás sorrendjével, egy munkafile (esetleg munkafile-ok) létrehozásával elkerülhetjük az adathalmaz tárbeli kezelését. (A ford. megjegyzése.)

Ez a táblázat egy olyan file tárbeli elhelyezését szemlélteti, amely 6, egyenként 3 mezőt tartalmazó rekordból áll. A D$ változót DIM D$(6,3) utasítással deklarálhatjuk.
A következő program a lemezen létrehoz egy ilyen szerkezetű file-t:

P. 9.

```
100 open 1,8,2,"tesztfile.seq,s,w"
110 for x=1 to 6
120 print chr$(147);
130 print"adatsorszam";x
140 print"---------------"
150 for y=1 to 3
160 print"mezoszam ";y;": ";
170 input x$
180 print#1,x$
190 next y
200 next x
210 close1
```

Az adathalmaz létrehozása kétszeresen egymásba ágyazott ciklussal történt, amelyben a ciklusváltozók adják meg a rekordok, azon belül a mezők sorszámát. A program lefutása után a 6 rekord soros file-ként helyezkedik el a lemezen. Célszerű ezt a programot SAVE "TEST. INP",8 utasítással tárolni a lemezen, a későbbiekben szükség lehet rá.
A file visszaolvasására szolgáló program szerkezete hasonló az előzőhöz, az eltérés mindössze annyi, hogy az egyes mezőket azonnal a táblázat, illetve az indexes változó megfelelő elemébe töltjük.

P. 10.

```
100 open 1,8,2,"tesztfile.seq,s,r"
110 dim d$(6,3)
120 for x=1 to 6
130 for y=1 to 3
140 input#1,d$(x,y)
150 next y
160 next x
180 close1
```

Végrehajtás után PRINT utasítással ellenőrizhetjük, hogy az adatokat szabályos helyen tároltuk-e.
Mivel az indexekkel minden mező címezhető, írjuk be például a PRINT D$ 1,2 utasítást az 1. rekord 2. mezőjének ellenőrzésére. Célszerű egy kiválasztott rekord összes mezőjét ellenőrizni. Tároljuk ismét a fenti programot, majd az alábbi rutinnal végezzük el ezt az ellenőrzést:

P. 11/a

```
100 input"a rekord sorszama:      ";x
120 print"----------------------------"
130 print"1. mezo: ";d$(x,1)
140 print"2. mezo: ";d$(x,2)
150 print"3. mezo: ";d$(x,3)
```

A program bekéri a rekord sorszámát, majd kiírja az összes mezőt úgy, hogy az első index a beírt változó, a második pedig felveszi az összes lehetséges értéket.
A táblázat tartalmát a tárban tetszőlegesen megváltoztathatjuk. Toldjuk be a fenti programba az alábbi sorokat:

P. 11/b

```
160 print"------------------------------"
170 input"modositando mezoszam:  ";y
180 input"uj tartalom            ";d$(x,y)
190 print"o.k"
200 print"tovabbi javitas (i/n)? "
210 get x$:if x$="" then 210
220 if x$="i" then 100
230 if x$="n" then end
240 goto 210
```

A megváltoztatandó mező sorszámát második indexként használjuk. A módosított táblázatot ismét tárolnunk kell a lemezen, erre alkalmas a következő rutin (előzőleg azonban tároljuk a módosított rutint!):

P. 12.

```
100 open 1,8,2,"@:tesztfile.seq,s,w"
110 for x=1 to 6
120 for y=1 to 3
130 print#1,d$(x,y)
140 next y
150 next x
160 close 1
```

Az egymásba ágyazott ciklus alkalmazásával ez a rutin is viszonylag rövid. A file-név előtt elengedhetetlen az "@:", miután a már meglévő régi file-t felül kell írni.
Az ilyen táblázatok használatánál rendkívül gyors az adatelérés. Az elérési idő nem függ a táblázat nagyságától, a táblázat nagysága viszont, s így az adatmennyiség is a tárkapacitás függvénye. Ha például a címkezeléshez egy olyan programot írunk, amely 8 kbyte-ot vesz igénybe, még mindig 30 kbyte marad a címek tárolásához. Miután egy cím tárolásához kb. 80 karakter szükséges, így nem kevesebb, mint 384 címet kezelhetünk a tárban! S mindezt olyan elérési idővel, amelyet a legbonyolultabb adatszervezéssel (index-szekvenciális) sem lehet túlszárnyalni.

1.4.8. Keresés a táblázatban

Mint a táblázatos feldolgozás kapcsán elmondottakból már tudjuk, egy táblázat bármelyik eleme indexelhető. A táblázat kétdimenziós, amelyben az első index adja meg a rekord sorszámát. Ha a tárba betöltünk egy ilyen táblázatos szerkezetű adathalmazt, majd abban meg akarunk keresni egy rekordot, ismernünk kell a keresett rekord sorszámát. Adott esetben ez a szám például a cikk- vagy vevőszám lehet. Vannak azonban olyan file-ok is, amelyek nem tartalmaznak megfelelő mezőt a rekordok sorszámozásához, tehát a keresést a mezők tartalma alapján kell elvégezni. Ehhez végig kell nézni a táblázatban lévő összes rekordot, majd egyenként összehasonlítani a rekordon belüli mezőt a keresettel. Nézzünk meg erre egy gyakorlati példát, de előbb az alábbi programmal állítsunk elő egy file-t, amelyben neveket és telefonszámokat tárolunk:

P. 13.

```
100 open 1,8,2,"teldat,s,w"
110 printchr$(147)
120 input"Vezeteknev        :";nn$
130 input"Keresztnev        :";vn$
140 input"Korzetszam        :";kn$
150 input"Allomasszam       :";an$
160 print"Rendben? (i/n)"
170 getx$:ifx$="" or x$<>"i" and x$<>"n" then170
180 ifx$="n"then110
190 print"Tovabbi bevitel? (i/n)"
200 getx$:ifx$="" or x$<>"i" and x$<>"n" then170
210 ifx$="n"then240
220 print#1,nn$",","vn$",","kn$",","an$
230 goto 110
240 close1
```

A program dokumentálása a következő:

100.	sor	a TELEDAT soros file megnyitása írásra
110.	sor	a képernyő törlése
120–150.	sorok	beolvassuk klaviatúráról a négy mezőt
160–180.	sorok	javítási lehetőséget adunk
190–210.	sorok	az adatbevitel és a program vége
220.	sor	a rekord négy mezőjét egymás után felírjuk a file-ba
230.	sor	folytatjuk az adatbevitelt
240.	sor	a 100. sorban megnyitott file-t lezárjuk.

Gépeljük be ezt a programot, indítsuk el és töltsük fel a file-t adatokkal. A tesztprogramot lemezen tároljuk azzal a céllal, hogy a munka végén a kész rutinokat egyetlen programmá fűzhessük össze. (A témakör utolsó fejezetében azonban megtalálhatjuk a telefonregiszterünk kezeléséhez szükséges teljes programot.)
A létrehozott adathalmazban többféle módszerrel végezhetünk keresést. Ha például egy név egy telefonszámot szeretnénk megtudni, kiírathatjuk az egész file-t képernyőre vagy nyomtatóra. Ez az eljárás meglehetősen időigényes, különösen akkor, ha a file sok rekordot tartalmaz.
Egy meghatározott név telefonszám szerinti megkeresését rábízhatjuk a gépre, amely egy ciklusban átfut az egész adathalmazon és megkeresi, majd kiírja a képernyőre vagy nyomtatóra azt a rekordot, amelyben a kívánt név szerepel:

P. 14.

```
100 open 1,8,2,"teledat,s,r"
110 dim d$(100,4):x=1
120 input#1,d$(i,1),d$(i,2),d$(i,3),d$(i,4)
130 ifst<>64then x=x+1:goto120
140 close1
150 printchr$(147)
160 input"a keresett nev: ";s$
170 fori=1tox
180 if d$(i,1)=s$ then210
190 nexti
200 print"Nincs ilyen nev":goto 280
210 print"A keresett nev:"
```

```
220 print"------------------"
230 print"Vezeteknev        :";d$(i,1)
240 print"Keresztnev        :";d$(i,2)
250 print"Korzetszam        :";d$(i,3)
260 print"Allomasszam       :";d$(i,4)
270 print"-------------------"
280 print"Tovabbi kereses? (i/n)"
290 getx$:ifx$="" or x$<>"i" and x$<>"n" then290
300 ifx$="i"then150
310 print"program vege...":end
```

A program dokumentálása:

100.	sor	a TELEDAT soros file megnyitása olvasásra
110.	sor	a rendszer a táblázatot 100 rekordra deklarálja és az indexet az 1-re állítja
120.	sor	a táblázat feltöltése
130.	sor	a file végét jelző státusz-változó (ST) ellenőrzése; ha nincs file-vég, az index 1-gyel nő
140.	sor	a file lezárása
150.	sor	képernyőtörlés
160.	sor	a keresett név beolvasása a billentyűzetről, tárolása az S$ változóban
170–190.	sorok	a ciklus a táblázatban azt a rekordot keresi, amelynek a névmezője megegyezik a keresett névvel. A rekord megtalálása után elágazás az output rutinhoz
200.	sor	nincs meg a név
210–270.	sorok	a keresett nevet tartalmazó rekord teljes kiírása
280–310.	sorok	lehetőség további keresésre

Láthatjuk, hogy ez a keresési folyamat nagyobb adatmennyiségnél is gyors, miután már a keresést megelőzően a file-t táblázatként töltöttük be a tárba. A táron belüli keresés gyorsabb a lemezen való keresésnél. A program könnyen módosítható úgy, hogy ne csak a név, hanem egy tetszőleges másik mező szerint is keressen. A fenti program a keresési szempontnak megfelelő első rekord megtalálása után megszakítja a keresést, ez azonban nem mindig célszerű. Ha például az előállított telefon-file-ban egy meghatározott lakcímnek megfelelő összes rekordot kell megkeresni és kiírni, egy másik rutinra van szükség. Szervezzük úgy a programot, hogy miután a rutin megtalált egy rekordot, annak kiírása után folytassa a keresést. A fenti követelményeknek az alábbi program felel meg:

P. 15.

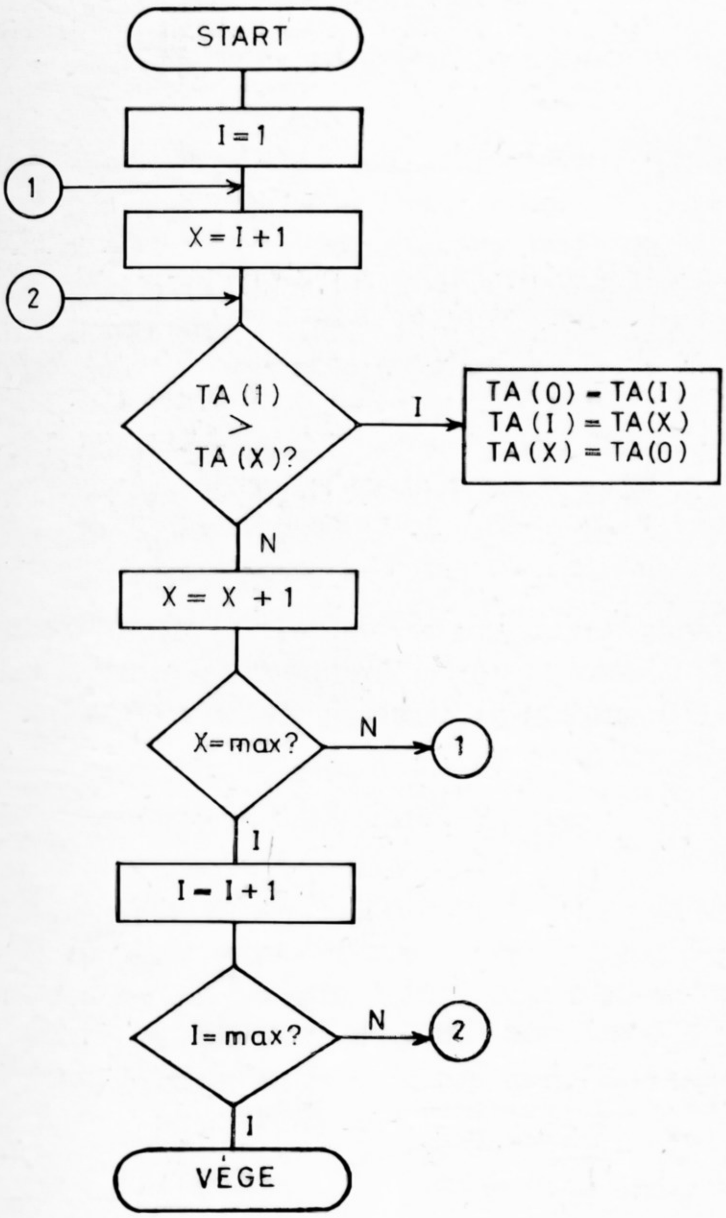
```
100 open 1,8,2,"teledat,s,r"
110 dim d$(100,4):x=1
120 input#1,d$(i,1),d$(i,2),d$(i,3),d$(i,4)
130 ifst<>64then x=x+1:goto120
140 close1
150 printchr$(147)
160 input"A keresett korzetszam:";s$
170 fori=1tox
180 if d$(i,1)=s$ then210
190 nexti
200 print"Adatsor vege!":goto 270
210 print"------------------"
220 print"Vezeteknev        :";d$(i,1)
230 print"Keresztnev        :";d$(i,2)
240 print"Korzetszam        :";d$(i,3)
```

```
250 print"Allomasszam      :";d$(i,4)
260 print"--------------------"
270 print"Tovabbi kereses? (i/n)"
280 getx$:ifx$="" or x$<>"i" and x$<>"n" then280
290 ifx$="i"then150
300 print"program vege...":end
```

A keresett rekord megtalálása és kiírása után nem fejeződik be a ciklus, a program a keresést addig folytatja, ameddig a felhasználó kívánja, illetve, amíg a file végét nem érzékelte. Ha az olvasó megértette ennek a programnak a működését, a vezetéknév szerint kereső programot önállóan is meg tudja írni, a fenti utasítássorozat igénybevételével.

1.4.9. Táblázat egyszerű rendezése

Az adatfeldolgozás során gyakran válik szükségessé egy adathalmaz numerikus vagy alfanumerikus átrendezése. Mivel a rendezés meglehetősen időigényes munka még a gép számára is, a programozók újabb és újabb rendezési eljárásokat dolgoznak ki a futási idő lerövidítése érdekében. A BASIC értelmező (interpreter) alatt írt rendezési eljárás az értelmező működési elvéből adódóan különösen sok időt vesz igénybe.



Miért kell tulajdonképpen rendezni az adatokat? Képzeljünk el egy olyan telefonkönyvet, amely a neveket rendezetlen sorban tartalmazza. Ahhoz, hogy egy bizonyos nevet megtaláljunk, elejétől végig át kell néznünk a telefonkönyvet. A rendezettség tehát előnyös egy adathalmazon belüli adatkeresésnél. A gép is jóval gyorsabban kiismeri magát egy rendezett fileban.

Több rendezési eljárás létezik, amelyek elsősorban a végrehajtási sebesség tekintetében térnek el egymástól. A legegyszerűbb rendezési eljárás az, amely egyenként a megfelelő helyre viszi az elemeket. Ha egy számsort emelkedő sorrendbe kell rendezni, az első elemet összehasonlítjuk a másodikkal. Ha az első a nagyobb, a program felcseréli az elemeket. Ezután az első és a harmadik elem összehasonlítására kerül sor, és így tovább egészen addig, míg az utolsó elemet elérjük. Ezután már a legkisebb elem került előre, azaz megfelelő helyre. A következő futás tehát már figyelmen kívül hagyhatja az első elemet. Ezt a logikát mutatja be az itt látható folyamatábra.

Ez a rendezőprogram az 1. indextől indul, amelyet kezdőindexként tárol az I változóban. A 2. index az X változó, amely eggyel nagyobb mint az I kezdőindex. Ezután következik az első és második elem összehasonlítása. Amennyiben a TA(I) tartalma nagyobb a TA(X) tartalmánál, ezeket az elemeket a TA(0) segédváltozón keresztül a programnak fel kell cserélnie egymással. Ezután az X index értéke eggyel nő, vagyis három lesz, majd a TA(I)-t, azaz ismét az első elemet hasonlítjuk össze a TA(X) harmadikkal és így tovább. Az utolsó elem elérésekor (X > utolsó index) az első helyen a legkisebb elem lesz és az I kezdőindexet eggyel növelhetjük. Ezután kezdődik elölről a ciklus.

A vázolt rendezési eljárás az első pillantásra elég körülményesnek tűnik. Az összehasonlításokat a tárban végezzük, viszonylag gyorsan. Kisebb rendezendő adathalmaznál ez a módszer nem rossz. Ha nem számsort, hanem karakteres adathalmazt akarunk abc sorrendbe rendezni, a programozástechnikai módszer megegyezhet az előzővel, ha figyelembe vesszük, hogy a BASIC hogyan végzi el két füzér összehasonlítását.

Töltsünk fel egy 12 elemű karakteres indexes változót tetszőleges szövegekkel:

P. 16/a

```
100 dim ta$(12)
110 for i=1 to 12
120 input ta$(i)
130 next i
```

Adatbevitel után a következő programrész abc-be rendezi a szövegeket, majd az eredményt kiírja képernyőre:

P. 16/b

```
140 i=1
150 x=i+1
160 if ta$(i) < ta$(x) then 180
170 ta$(0)=ta$(i):ta$(i)=ta$(x):ta$(x)=ta$(0)
180 x=x+1
190 if x <= 12 then 160
200 i=i+1
210 if i <> 12 then 150
220 for i=1 to 12
230 print ta$(i)
240 next i
```

Ha névsor helyett teljes rekordokat, azaz egy egész táblázatot kell rendezni a tárban (pl. az előző telefon-file-unkat), rendezés közben az összes mezőt, azaz egész rekordokat kell felcserélni. A név szerinti rendezéshez a 160–170. sorokat az alábbiak szerint módosítjuk:

P. 17

```
100 dim ta$(12,3)
110 for i=1 to 12
120 input ta$(i,1),ta$(i,2),ta$(i,3)
130 next i
140 i=1
150 x=i+1
160 if ta$(i,1) < ta$(x,1) then 180
170 ta$(0,1)=ta$(i,1):ta$(i,1)=ta$(x,1):ta$(x,1)=ta$(0,1)
171 ta$(0,2)=ta$(i,2):ta$(i,2)=ta$(x,2):ta$(x,2)=ta$(0,2)
172 ta$(0,3)=ta$(i,3):ta$(i,3)=ta$(x,3):ta$(x,3)=ta$(0,3)
180 x=x+1
190 if x <= 12 then 160
200 i=i+1
```

```
210 if i <> 12 then 150
220 for i=1 to 12
230 print ta$(i)
240 next i
```

Nagyobb adatállomány ilyen módon történő rendezése rendkívül időigényes. Ha gyors rendezést akarunk végezni, használhatjuk pl. a DATA BECKER által kiadott "Tippek és trükkök a C 64-en"* c. könyvben levő rendkívül gyors gépi kódú rendező rutint.

1.4.10. Címkezelés soros adattárolással

A fejezet végén közlünk egy kényelmes címkezelő programot, amelyet valószínűleg minden olvasó célszerűen fel tud használni. Ez a program különböző jellegű adatállományok feldolgozására alkalmas.
A programban használt címrekord szerkezete:

- NÉV 1.
- NÉV 2.
- UTCA/HÁZSZÁM
- IRÁNYÍTÓSZÁM/HELYSÉG
- TELEFON
- MEGJEGYZÉS

A NÉV 1. és a NÉV 2. mező használata a felhasználó belátására van bízva. Tárolhatjuk a NÉV 1.-ben a vezetéknevet, a NÉV 2.-ben pedig a keresztnevet, vagy az egyik mezőben a cég nevét, a másikban a címzett személy nevét stb.
A MEGJEGYZÉS mező például a címek csoportosításához használható (családtagok, hivatali munkatársak, barátok stb.).
Az adathalmazt kezelő program menüje az alábbi lehetőségeket tartalmazza:

–1– ADATHALMAZ BETÖLTÉSE
–2– ADATHALMAZ MENTÉSE
–3– ADATBEVITEL
–4– ADATJAVÍTÁS
–5– KERESÉS/LISTÁZÁS
–6– ADATOK TÖRLÉSE
–0– PROGRAM BEFEJEZÉSE

–1– AZ ADATHALMAZ BETÖLTÉSE

Ez a rutin megkérdezi a felhasználótól az adathalmaz nevét, majd ha ilyen nevű file létezik a lemezen, annak tartalmát beolvassa a tárba. Ezután a képernyőn megjelenik a file által tartalmazott rekordok száma. Ha beolvasás során hiba lép fel, vagy a file egyáltalán nem létezik, a rutin a LEMEZHIBA üzenetet írja ki. A rutin RETURN-nel történő befejezése után ismét megjelenik a menü.

* 64 Tips und tricks (DATA BECKER, 1984), magyarul előkészületben (a szerk. megjegyzése).

–2– ADATHALMAZ MENTÉSE

Ha az adathalmazt betöltés után megváltoztattuk vagy bővítettük, a program befejezése előtt ezzel a rutinnal a módosított állományt ki kell írnunk a lemezre. A mentés során a file ugyanazon a néven kerül fel a lemezre, amilyen néven eddig szerepelt (vagyis amit a betöltés rutinban a felhasználó megadott).
Ezt a rutint célszerű a feldolgozás során többször behívni, előfordulhat ugyanis pl. egy rövid áramszünet, amikor nyilván az összes aktuális módosításunk elveszhet. Mentés után nyugodtan dolgozhatunk tovább a tárban lévő adathalmazzal.

–3– ADATBEVITEL

Ennek a rutinnak két funkciója van:

1. Ha még üres az adathalmaz, ezzel a rutinnal létrehozhatjuk, majd feltölthetjük adatokkal. A felhasználónak meg kell adnia a file nevét, lehetőleg ügyelve arra, hogy a lemezen ilyen nevű állomány még ne szerepeljen, azt ugyanis a mentést végző rutin értelemszerűen felülírja.
2. Ha már vannak adatok a tárban, a rutin elvégzi az adathalmaz bővítését. Minden rekord bevitele után megjelenik a HELYES(I/N)? üzenet. Itt lehetőség nyílik a beírt adatok helyesbítésére. Ha az összes adatot szabályosan írtuk be, úgy az I-t kell leütni. Most megjelenik a TOVÁBBI BEVITEL(I/N)? üzenet. Ha az N billentyűt nyomjuk meg, a program visszatér a menühöz.

–4– ADATJAVÍTÁS

A módosító rutinnal bármely rekord mezőit megváltoztathatjuk. Valamilyen módon azonban közölni kell a géppel, hogy melyik a javítandó rekord.
A program először bekéri a felhasználótól a keresett rekordban szereplő két nevet és azt a rekordot írja ki módosításra, amelyben ezek a nevek szerepelnek. Módosítás előtt tájékozódhatunk az adathalmazban a KERESŐ/LISTÁZÓ rutin segítségével.
Miután a gép kiírta a javítandó rekordot, meg kell adnunk a módosítandó mező számát. Ezt követi az új tartalom meghatározása. A címet még egyszer kiírja a gép módosított formában. Ha ebben a rekordban további módosítások nem szükségesek, a 9 billentyűvel fejezhetjük be a rekord javítását. Végül a program megkérdezi, akarunk-e további rekordokat javítani. Ezt a kérdést az I és az N billentyűvel válaszolhatjuk meg.

–5– KERESÉS/LISTÁZÁS

Ez a rutin rendkívül komplex és sokoldalú. Kérdés alapján választhatunk, hogy a keresett címeket a képernyőre (B billentyű) vagy a nyomtatóra (D billentyű) akarjuk listázni. Ha a nyomtató mellett döntöttünk, még egyszer kell választanunk az alábbi két lehetőség közül:

- a címek kinyomtatása az összes mezővel közönséges nyomtatópapíron
- az 1–5. mezők kinyomtatása felragasztható címkéken.

Az első esetben a P, a másodikban az A billentyűt kell leütni. A felragasztható cím-címkék egysorosak és 89 × 36 mm formátumúak lesznek.

Az adatok válogatásához ki kell töltenünk egy ún. *keresési maszk*ot. Ha egy mező nem meghatározott a keresésnél, a RETURN-t kell leütni. Ha például az összes olyan címet keressük, amelyben az irányítószám 4000, minden mezőnél üres RETURN-t kell ütni kivéve az irányítószámot. Az IRÁNYÍTÓSZÁM/HELYSÉG mezőben a 4-es számot adjuk meg, majd utána a RETURN-t.

Néhány példa a válogatásra:

VEZETEKNEV	RETURN
KERESZTNEV	RETURN
UTCA/HAZSZAM	RETURN
IRANYITOSZAM/HELYSEG	4000
TELEFON	RETURN
MEGJEGYZES	RETURN

A fenti válogatással minden 4000-es irányítószámú rekordot kiír a gép.

VEZETEKNEV	M
KERESZTNEV	RETURN
UTCA/HAZSZAM	RETURN
IRANYITOSZAM/HELYSEG	RETURN
MEGJEGYZES	CSALAD

Az utóbbi esetben minden olyan családtag címét kiírja a gép, ahol a VEZETEKNEV mező első betűje M.

–6– ADATOK TÖRLÉSE

A VEZETÉKNÉV és a KERESZTNÉV beírása után a gép kiírja a neveket és megkérdezi, hogy ezt a címet valóban törölni kell-e. A törlésre csak az I billentyű megnyomása után kerül sor.

–0– A PROGRAM BEFEJEZÉSE

Előfordulhat, hogy a menütől a program végét kérjük, de előtte elfelejtettük lemezre vinni (menteni) az adatállományt. Ekkor GOTO 110-zel újra elindíthatjuk a programot anélkül, hogy az adatok törlődnének;

Nézzük át a program listáját:

P. 18.

```
100 poke 53280,5:poke53281,2:printchr$(158);:dimd$(100,7)
110 gosub2030
120 print"Valassza ki a funkciot:"
130 print"----------------------------------------":print
140 print"          -1- adathalmaz betoltese
150 print"          -2- adathalmaz mentese
160 print"          -3- adat bevitel
170 print"          -4- adat javitas
```

```
180 print"          -5- adat kereses,listazas"
190 print"          -6- adat torles":print
200 print"          -0- program befejezese"
210 print
220 print"          valasszon  (0-6)?"
230 getx$:ifx$<"0"orx$>"6"then230
240 ifx$<>"0"then340
250 print:print"          biztos   (i/n)?"
260 getx$:ifx$<>"n"andx$<>"i"then260
270 ifx$="n"then110
280 gosub2030
290 printtab(3);"A program ujra indithato"
300 printtab(15);"'goto 110'":print
310 printtab(3);"utasitassal - adatelvesztes nelkul!!":print
330 end
340 onval(x$)gosub360,540,680,880,1190,1770
350 goto110
360 rem ************
370 rem adat olvasas
380 rem ************
390 gosub2030
400 input"Az adatfile neve:";dn$
410 open15,8,15
420 open1,8,2,dn$+",s,r"
430 input#15,fe:if fe=0then460
440 print"disk hiba !!!"
450 goto510
460 x=1
470 input#1,d$(x,1),d$(x,2),d$(x,3),d$(x,4),d$(x,5),d$(x,6),d$(x,7)
480 if st<>64 then x=x+1:goto 470
490 print"A filebol osszesen";x
500 print"rekord kerult beolvasasra.:print
510 close1:close15
520 print"tovabb - [RETURN]"
530 inputx$:return
540 rem ************
550 rem    mentes
560 rem ************
570 if x>0then590
580 gosub2230:return
590 gosub2030
600 open1,8,2,"@:"+dn$+",s,w"
610 fori=1tox
620 print#1,d$(i,1)","d$(i,2)","d$(i,3)",";
630 print#1,d$(i,4)","d$(i,5)","d$(i,6)","d$(i,7)
640 next
650 print"Adatok kimentve!":close1:print
660 print"tovabb - [RETURN]"
670 inputx$:return
680 rem ***********
690 rem adatbevitel
700 rem ***********
710 ifx>0thengoto730
720 gosub2030:input"Az adatfile neve:";dn$
730 x=x+1
740 gosub2030
750 print"Adat bevitel:"
760 print"-------------":print
770 i=x:gosub2110
780 fori=1to7:printchr$(145);:next
790 fori=1to7:printtab(15);:inputd$(x,i):next
800 print:print"rendben (i/n)?"
810 getx$:ifx$<>"n"andx$<>"i"then810
820 ifx$="i"then840
830 goto740
840 print"Tovabbi bevitel (i/n)?"
850 getx$:ifx$<>"i"andx$<>"n"then850
```

```
860 ifx$="i"then730
870 return
880 rem ***********
890 rem adat javitas
900 rem ************
910 ifx>0then930
920 gosub2230:return
930 gosub2030
940 input"Vezeteknev: ";n1$
950 input"Keresztnev: ";n2$
960 fori=1tox
970 ifd$(i,1)=n1$andd$(i,2)=n2$then1010
980 nexti
990 print"Ilyen nev nincs!"
1000 print"tovabb - [RETURN]":inputx$:return
1010 gosub2030
1020 print"-1- vezeteknev    :";d$(i,1)
1030 print"-2- keresztnev    :";d$(i,2)
1040 print"-3- lakcim - utca :";d$(i,3)
1050 print"-4-          - varos:";d$(i,4)
1060 print"-5- iranyitoszam  :";d$(i,5)
1070 print"-6- telefonszam   :";d$(i,6)
1080 print"-7- megjegyzes    :";d$(i,7):print:print
1090 print"Kerem valasszon: ":print"(9=nem modosit)":print
1100 getx$:ifval(x$)<1 or val(x$)>7andval(x$)<>9then1100
1110 ifval(x$)=9then1150
1120 y=val(x$)
1130 input"Uj adattartalom:";d$(i,y):print
1140 goto1010
1150 print"Tovabbi javitas (i/n)?"
1160 getx$:ifx$<>"i"andx$<>"n"then1160
1170 ifx$="i"then880
1180 return
1190 rem ************************
1200 rem adat valasztas - listazas
1210 rem ************************
1220 ifx>0then1240
1230 gosub2230:return
1240 gosub2030:print"Lista printerre (p),vagy kepernyore (s)?";
1250 getx$:ifx$<>"s"andx$<>"p"then1250
1260 o$=x$:ifo$="s"then1300
1270 print:print"papir (p), vagy level cimke (m)?"
1280 getx$:ifx$<>"p"andx$<>"m"then1280
1290 d$=x$
1300 gosub2030
1310 print"A keresesi kulcsot kerem."
1320 print"Ha nincs feltetel - csak [RETURN]"
1330 print"---------------------------------":print
1340 i=0:gosub2110
1350 fori=1to7:printchr$(145);:s$(i)="":next
1360 fori=1to7:printtab(15);:inputs$(i):next
1370 ifo$="s" or d$="m"then1450
1380 gosub2030:print"A printer rendben (i)?"
1390 getx$:ifx$<>"i"then1390
1400 open1,4
1410 print#1,"Nev  1";spc(8);"Nev  2";spc(9);"utca  ";spc(10);
1420 print#1,"varos":print#1,"ir.szam    tavb.sz.  megj."
1430 fori=1to79:print#1,"=";:next:print#1
1440 close1
1450 fori=1tox
1460 fory=1to7
1470 ifs$(y)=left$(d$(i,y),len(s$(y)))thenz=z+1:goto1480
1480 nexty
1490 ifz=7then gosub 1550
1500 z=0:nexti
1510 print:print"Adatsor vege!":print
1520 print"tovabb - [RETURN]":print
```

```
1530 inputx$
1540 return
1550 ifo$="s"then1730
1560 ifd$="m"then1670
1570 open1,4
1580 print#1,d$(i,1);spc(10-len(d$(i,1)));
1590 print#1,d$(i,2);spc(20-len(d$(i,2)));
1600 print#1,d$(i,3);spc(20-len(d$(i,3)));
1610 print#1,d$(i,4)
1620 print#1,spc(3);d$(i,5);spc(25-len(d$(i,5)));
1630 print#1,d$(i,6);spc(15-len(d$(i,6)));
1640 print#1,d$(i,7)
1650 print#1:close1
1660 return
1670 open2,4
1680 print#2
1690 forj=1to5:print#2,d$(i,j):next
1700 print#2:print#2:print#2
1710 close2
1720 return
1730 gosub2030:gosub2110
1740 print:print"tovabb (i)?"
1750 getx$:ifx$<>"i"then1750
1760 return
1770 rem ***********
1780 rem adat torles
1790 rem ***********
1800 ifx>0then1820
1810 gosub 2230:return
1820 gosub2030
1830 input"Vezeteknev   : ";n1$
1840 input"Keresztnev   : ";n2$
1850 fori=1tox
1860 ifd$(i,1)=n1$andd$(i,2)=n2$then1900
1870 nexti
1880 print"Ilyen nev nincs!":print
1890 print"tovabb - [RETURN]":inputx$:return
1900 gosub2030:gosub2110
1910 print:print"Torli a rekordot (i/n)?"
1920 getx$:ifx$<>"i"andx$<>"n"then1920
1930 ifx$="n"thenreturn
1940 fory=itox-1
1950 forj=1to7
1960 d$(y,j)=d$(y+1,j)
1970 nextj,y
1980 forj=1to7:d$(x,j)="":nextj
1990 x=x-1
2000 print"A rekord torolve!"
2010 print"tovabb - [RETURN]"
2020 inputx$:return
2030 rem ***********
2040 rem program fej
2050 rem ***********
2060 printchr$(147);
2070 printtab(8);"======================="
2080 printtab(8);"C I M   nyilvantartas
2090 printtab(8);"=======================":print:print
2100 return
2110 rem ***********
2120 rem rekord iras
2130 rem ***********
2140 print"Vezeteknev    : ";d$(i,1)
2150 print"Keresztnev    : ";d$(i,2)
2160 print"Lakcim - utca : ";d$(i,3)
2170 print"         varos: ";d$(i,4)
2180 print"Iranyitoszam  : ";d$(i,5)
2200 print"Telefonszam   : ";d$(i,6)
```

```
2210 print"Megjegyzes       : ";d$(i,7)
2220 return
2230 rem ************
2240 rem nincs adat !
2250 rem ************
2260 gosub2030
2270 print"Nincs adat a memoriaban!":print
2280 print"tovabb - [RETURN]"
2290 inputx$:return
```

1.4.11. A soros adattárolás alkalmazási területei

A következő fejezetekben ismertetendő relatív- és közvetlen elérésű file-okhoz viszonyítva a soros file egyik előnye a lemezterülettel való rendkívül takarékos bánásmód. A legkülönbözőbb hosszúságú adatok tárolhatók folyamatosan egymás után anélkül, hogy ehhez előre rögzítenünk kellene a rekordhosszúságot.
Ezzel a lehetőséggel élhetünk olyan esetben, amikor az adathalmaz részeit nem kell megváltoztatni, s amikor a rekordokat nem kell közvetlenül elérni. Az idevonatkozó példák a következők:

Könyvelői file-ok
Egy könyvelési napló folyamatosan ellenőrzi az összes könyvelési folyamatot. Változtatásokra nincs mód.
Kiértékelési file-ok
Egy közvetlen elérésű file feldolgozása során például az 5000 Ft-nál nagyobb értékben vásároló összes vevőt, és a további nyomtatáshoz a megtalált rekordokat beírhatjuk egy soros file-ba.

Ha az olvasó még nem rendelkezik elegendő programozási tapasztalattal, soros file segítségével szimulálhatja a közvetlen elérésű feldolgozást (a tárban dolgozva). Ajánlatos azonban az adattárolás egyéb lehetőségeit is áttanulmányozni, mert ezzel saját munkánkat könnyítjük meg.

1.5. A relatív adattárolás

A relatív adattárolást és ennek programozását a VC–1541-es felhasználói kézikönyve nem ismerteti. Ennek magyarázata, hogy a VC–20-as és a COMMODORE 64-es BASIC 2.0 programozási nyelve nem tartalmazza a relatív file-ok kezelésével kapcsolatos utasításokat. Így elvileg a CBM 64-gyel és a VC–20-szal nincs lehetőség relatív adattárolásra – de csak elvileg. A gyártó ugyanis kifejlesztett néhány mesterfogást, amelyekkel megkerülhetjük a BASIC 2.0 korlátait, s a relatív adattárolást a VC–20-as és a COMMODORE 64-es gépeken is használhatjuk. Előfordulhat egyes esetekben, hogy ez kissé nehézkesnek tűnik – például amikor a rekordhosszúsággal kapcsolatos adatokat CHR$(x)kóddal kell átvinni a lemezegységhez – azonban a kényelmesebb adatkezelésért érdemes ennyi fáradtságot vállalni.

1.5.1. Az elv

A relatív file azonos hosszúságú, sorszámmal ellátott rekordokkal dolgozik. A sorszám alapján minden rekord közvetlenül címezhető, azaz ha megadjuk a rekordszámot, a rekordot

közvetlenül beolvashatjuk. A továbbiakban a rekord sorszámát *kulcsnak* nevezzük. A kulcs alapján a DOS felismeri, hogy a file kezdetéhez „viszonyítva" a rekord hol helyezkedik el a lemezen, így az olvasófejet erre a helyre pozicionálva a rekordot el tudja olvasni.
Tehát nem kell beolvasni a teljes adathalmazt vagy indextáblázatot a tárba, hanem csak az éppen szükséges rekordokat.

A relatív file kezelésének lépései

A) A relatív file létrehozása:
1. a file-t megnyitjuk. Ekkor történik a rekordok hosszúságának meghatározása.
2. az utolsó rekord deklarálása
3. a file-t lezárjuk

B) Egy rekord írása a file-ba:
1. a file megnyitása
2. pozicionálás a kívánt rekordra
3. a rekord kiírása
4. a file lezárása

C) Egy rekord olvasása a file-ból:
1. a file megnyitása
2. pozicionálás a kívánt rekordra
3. a rekord olvasása
4. a file lezárása.

Ez természetesen csak egy durva áttekintés. A következő fejezetekben az említett folyamatok részletes elemzése következik.

1.5.2. A relatív adattárolás előnyei

– a rekordokat gyorsabban elérjük
– tehermentesítjük a gép tárát

A soros file-oknál lényeges feltétel volt, hogy az adatkezeléshez a soros file-nak a tárban kellett lennie. Ellenkező esetben ugyanis egy rekord keresésekor az egész file-t végig kellett volna olvasnunk. Ez gyakorlatilag annyit jelent, hogy az összes rekord beolvasásán túl a rekordokat a keresési kulccsal is össze kellene hasonlítani. A keresés ezzel az eljárással időigényessége miatt teljes képtelenség lenne.
A relatív file-oknál mindez jóval egyszerűbb. Mivel a kulcs segítségével minden rekordot közvetlenül beolvashatunk, a file mérete független a tárkapacitástól. Ez azt jelenti, hogy egy olyan programmal, amely a VC–20-as 3,5 kbyte-ját teljesen lefoglalja, egy max. 163 kbyte-nyi területet elfoglaló file-t tudunk kezelni.
Az a programozó, aki egyszer megismerte a relatív file-szervezés előnyeit, a továbbiakban ha csak teheti, egészen biztosan ezt az adatszervezési módot használja. Itt mutatkoznak meg igazán a relatív file-ban rejlő lehetőségek.

1.5.3. A relatív file megnyitása

A relatív file-okat is OPEN utasítással nyitjuk meg. Ez az utasítás alig tér el a soros file-ok megnyitásához szükséges utasítástól. A relatív file megnyitásának formátuma:

OPEN lfn,ga,csatorna,"file-név,L," + CHR$ (rekordhosszúság)

Az első négy paraméter azonos a soros file OPEN utasításában szereplő paraméterekkel. Azaz a paraméterek sorrendben:

a logikai file-szám,
az egységszám (általában 8),
a csatornaszám (2–14),
a file neve.

Az L betű közli a DOS-szal, hogy relatív file-t nyitottunk meg, amelyben a rekordok hossza a CHR$ argumentuma. A rekordhossz 1 és 254 között lehet, azaz egy rekord legfeljebb 254 karaktert tartalmazhat. Ha a rekordhosszúság 88-nál kisebb, a rekord egy INPUT utasítással olvasható. Az INPUT utasítás csak akkor működik helyesen, ha a PRINT utasítás a rekordot egy lezáró RETURN-nel együtt írja a lemezre, tehát ha a PRINT utasítást követő változó mögé nem írtunk pontosvesszőt. A RETURN a rekordoknak elengedhetetlen része, ha tehát a rekordokat INPUT-tal akarjuk beolvasni, az OPEN utasításban a tényleges rekordhossznál mindig 1-gyel nagyobb értéket kell megadnunk. Ezek szerint egy 80 karakteres rekordokból álló file-t a következőképpen kell megnyitni:

OPEN1,8,2,"FILE.REL,L" + CHR$(81)

A FILE.REL nevű relatív file megnyitása a 2. csatornán keresztül történt. A rekord teljes hossza a RETURN-nel együtt 81 karakter.

A rendelkezésre álló csatornák korlátozott száma miatt egy lemezegységen egyszerre csak egy relatív file-t nyithatunk meg. Ha két relatív file-lal akarunk dolgozni, az egyiket mindig le kell zárni, míg a másikkal dolgozunk.
A relatív file létrehozásakor célszerű a teljes file-területet előkészíteni, mert így a későbbi feldolgozás sokkal gyorsabbá válik. Az előkészítés azt jelenti, hogy a file legnagyobb kulcsú (sorszámú) rekordjába CHR$(255) karaktert írunk. Ha olyan rekord olvasásával próbálkozunk, amelynek a kulcsa nagyobb a file eddigi utolsó rekordjának kulcsánál, ez a RECORD NOT PRESENT hibához vezet. Ha azonban ugyanilyen kulcsú rekordot *írunk* a file-ba, a DOS az új rekord kulcsánál kisebb kulcsú összes rekordot automatikusan teleírja CHR$(255) karakterekkel. Az említett tartomány rekordjainak további olvasása ekkor hibamentesen valósul meg.
Az olyan rekordokat, amelyekben CHR$(255) van, *felszabadított rekord*oknak nevezzük. Ha a file összes rekordja fel van szabadítva, a feldolgozás sokkal gyorsabb, mint egyébként. Nézzünk erre egy példát:

Tegyük fel, hogy megnyitunk egy 100 rekordra tervezett relatív file-t, de az utolsó (100.) rekordot nem szabadítjuk fel. Ha most például először az 50-es kulcsú rekordot írjuk fel a file-ba, a DOS az első 49 rekordot automatikusan felszabadítja, azaz teleírja CHR$(255) karakterekkel, ez pedig elég soká tart. Ha a file legelső megnyitásakor a tervezett legnagyobb

kulcsú rekordba írunk egy CHR$(255) karaktert, a file összes rekordja felszabadul, azaz előkészítettük a lemezterületet. Így további feldolgozásunkat nem lassítja a rekordok felszabadításának művelete. A relatív file kezelésének legfontosabb része egy adott kulcsú rekord beolvasása és felírása, minthogy most a rekordokat nem folyamatosan olvassuk és írjuk, hanem közvetlenül a kiválasztott rekordra pozicionálunk. A pozicionálás fizikailag is lezajlik, hiszen a DOS a rekord kulcsából meghatározza azt a pozíciót, ahol a rekord a lemezen fizikailag elhelyezkedik és az író/olvasófejet az adott lemezterületre irányítja. A pozicionálásra szolgáló utasítás formátuma:

PRINT # lfn,"P" + CHR$(csatorna) + CHR$(alsó) + CHR$(felső) + CHR$(byte)

A lemezterület előkészítésekor még egyetlen rekord sincs felszabadítva, hiszen éppen most akarjuk ezt megtenni. A tervezett legnagyobb kulcsú rekordba írunk egy CHR$(255) karaktert. Írás előtt kiadjuk a fenti pozicionáló utasítást, ekkor az egységen a LED villogni kezd és a RECORD NOT PRESENT hibaüzenetet olvashatjuk le a hibacsatornáról. Ez azonban nem hiba, a rekord felírását a DOS elvégzi. A látszólagos hiba minden olyan esetben fellép, amikor az eddig felszabadított legnagyobb kulcsú rekordnál nagyobb kulccsal írunk fel egy rekordot. Felírás után a LED abbahagyja a villogást, a hibajelenség megszűnik. A pozicionáló utasításban az alsó és felső paraméterek adják meg a rekordok kulcsát. Miután az egy byte-ban ábrázolható legnagyobb szám 255, s egy relatív file maximum 65535 rekordot tartalmazhat, a kulcsot két byte-ban kell megadni. Ez a két byte az alábbi képlet segítségével számítható ki:

HB = INT(RN/256)
LB = RN–HB*256, ahol

HB = High Byte (felső byte)
LB = Low Byte (alsó byte)
RN = Record Number, a rekordok kulcsa (sorszáma).
Az utolsó paraméter a megadott rekordon belül egy meghatározott karakterre történő pozicionálásra szolgál.

PRINT # 2,"P" + CHR$(2) + CHR$(10) + CHR$(1) + CHR$(5)

A fenti pozicionálás a 266. rekord 5. byte-jára történik. A 266-os kulcsot a 10 (alsó byte) és 1 (felső byte) paraméterek határozzák meg.
A teljes rekord olvasásához vagy írásához a 1. karakterre (byte-ra) kell pozicionálnunk. Ha az utolsó paraméter nincs megadva, a pozicionálás a lezáró RETURN-re (CHR$(13)) történik. Egy egyenként 80 karakterből álló 1000 rekord hosszúságú file előkészítéséhez szükséges BASIC-készlet a következő:

P.19.

```
100 rn=1000
110 hb=int(rn/256)
120 lb=rn-hb*256
130 open1,8,2,"file.rel,l,"+chr$(80)
140 open2,8,15
150 print#2,"p"+chr$(2)+chr$(lb)+chr$(hb)+chr$(1)
160 print#1,chr$(255)
170 close1:close2
```

Az 1000 rekord felszabadítása kis időt vesz igénybe. Ne feledkezzünk meg arról, hogy ezekben a 80 karakteres rekordokban csak 79 karakteres rekordok helyezhetőek el.

1.5.4. Adatelőkészítés relatív tároláshoz

A relatív tároláshoz elengedhetetlen a fix rekordhosszúság. Ha a rekord több mezőből áll, a mezőket össze kell kapcsolni arra ügyelve, hogy az egész file-ban minden rekordon belül az egyes mezők egy és ugyanazon pozíción kezdődjenek.
Legjobb, ha ezt a problémát egy példán keresztül vizsgáljuk meg. Tegyük fel, hogy anyagnyilvántartást akarunk készíteni, amely a következő szerkezetű rekordokból áll:

CIKKSZÁM	4 helyértékű
RAKTÁRI SZÁM	5 helyértékű
BESZERZÉSI ÁR	6 helyértékű
MEGNEVEZÉS	15 helyértékű
ELADÁSI ÁR	6 helyértékű
Rekordhossz	36 byte

A cikkcsalád 36 byte rekordhosszúsággal kb. 200 cikket foglal magában. Hozzuk létre ezt a file-t:

P.20.

```
100 rn=200:rem a rekordok varhato max. szama
110 rl=36:rem a rekordok hossza
120 hb=int(rn/256)
130 lb=rn-hb*256
140 open1,8,2,"file.rel,l,"+chr$(rl)
150 open2,8,15
160 print#2,"p"+chr$(2)+chr$(lb)+chr$(hb)+chr$(1)
170 print#1,chr$(255)
180 close1:close2
```

Tegyük fel, hogy a cikkfile sorosan van megadva, 200 rekordból tevődik össze, amelyek egyenként 5 mezőből állnak. Ezeket a mezőket össze kell kapcsolni fix hosszúságú mezőkből álló rekordokká és felírni a relatív file-ba. A megoldásnál gondot okozhat, hogy pl. a cikk megnevezése nem mindig tölti ki a 15 karaktert. A relatív file szerkezetének a következőnek kell lennie:

POZÍCIÓ	111111111122222222223333333 123456789012345678901234567890123456				
MEZŐ	AN$	BE$	LN$	EP$	VP$
TARTALOM	1	2 mm LEMEZ	1344	23.40	42.30
	2	3 mm CSAVAR	1231	9.00	14.00
	3	A3A4 SZELEP	1243	23.45	29.90
	.	.	.	.	.
	200	12 mm TÖMLŐ	2321	6.70	9.80

A soros file mezőit az alábbi változókba olvassuk be:

Cikkszám	AN$
Megnevezés	BE$
Raktári szám	LN$
Beszerzési ár	EP$
Eladási ár	VP$

Bár a következő utasításban ezeket a mezőket összekapcsoljuk, de amint látni fogjuk, nem várt eredménnyel:

```
RC$=AN$+BE$+LN$+EP$+VP$
```

Az RC$ rekord nem felel meg a kívánt file-szerkezetnek. Ez azzal magyarázható, hogy a cikk megnevezése után közvetlenül a raktári szám következik. Miután azonban a raktári számnak okvetlenül a 20-as helyértéktől kell kezdődnie, holott a cikk megnevezése nem állandóan 15 karakter, a raktári szám mező „elcsúszik". Ahhoz, hogy a rekordokból visszaolvasás során az egyes mezőket egyértelműen visszanyerjük, elengedhetetlen a rekordszerkezet feltétel nélküli betartása. Azokat a mezőket, melyek hossza rövidebb a tervezettnél, a rekord összeállítása során üres jelekkel kell kiegészítenünk. Mindezek figyelembevételével a láncolás az alábbi formát ölti:

```
BL$="               "
RC$=AN$+LEFT$(BL$4-LEN(AN$))
RC$=RC$+BE$+LEFT$(BL$,15-LEN(BE$))
RC$=RC$+LN$+LEFT$(BL$,5-LEN(LN$))
RC$=RC$+EP$+LEFT$(BL$,6-LEN(EP$))
RC$=RC$+VP$+LEFT$(BL$,6-LEN(EP$))
```

Az összekapcsolás bonyolultabbnak tűnik, mint amilyen a valóságban. Minden egyes mezőt annyi üres karakterrel (space) kell kiegészíteni, amennyi a mező tervezett és tényleges hosszának különbsége. Ezeket az üres jeleket a program elején definiált BL$ füzér (string) szolgáltatja. Az említett füzér olyan hosszú, mint a rekord leghosszabb mezője, vagyis a mi esetünkben 15 karakter.
Nézzünk meg részleteiben egy példát. Tegyük fel, hogy az első cikkszám 8. Ha a mező maximális hosszából (4) elvonjuk a tényleges hosszúságot (1), 3-at kapunk. Az AN$ füzért tehát három üres (LEFT$(BL$,3)) jellel kell kiegészíteni.
Az adatokat a relatív file-ba történő átvitel előtt a fenti eljárással elő kell készíteni, ezért indokolt, hogy saját munkánk megkönnyítésére a relatív file-kezelő programokba egy olyan szubrutint írjunk, amely az egyes mezőket a tervezett hosszúságig kitölti üres jelekkel.

1.5.5. A lemezegység és a gép közötti adatátvitel

Ez az adatátvitel elvileg semmiben sem tér el a soros tárolásnál alkalmazott módszertől. A tételeket a PRINT-tel írjuk, s az INPUT-tal vagy a GET-tel olvassuk. Az egyetlen különbség az, hogy egy rekord olvasása vagy írása előtt pozicionálnunk kell az adott rekordra. Ez a P utasítással történik. Készítsünk most az alábbi programmal egy relatív file-t:

P.21.

```
100 bl$="               "
105 open1,8,2,"test.rel,l,"+chr$(41)
110 open2,8,15
120 print#2,"p"+chr$(2)+chr$(100)+chr$(0)+chr$(1)
130 print#1,chr$(255)
140 printchr$(147)
150 print"Adat   bevitel"
160 print"---------------"
170 input"Rekordszam (1-100)  : ";rn
180 if rn<1 or rn>100 then print chr$(145);:goto 160
190 input"Mezo 1 (max. 10 karakter): ";f1$
200 if len(f1$)>10 then printchr$(145);:goto190
210 input"Mezo 2 (max.  5 karakter): ";f2$
220 if len(f1$)> 5 then printchr$(145);:goto210
230 input"Mezo 3 (max. 10 karakter): ";f3$
240 if len(f1$)>10 then printchr$(145);:goto230
250 input"Mezo 4 (max. 15 karakter): ";f4$
260 if len(f1$)>15 then printchr$(145);:goto250
270 print"Rendben (i/n)?"
280 getx$:if x$<>"i" and x$<>"n" then280
290 if x$="n" then 140
300 rc$=f1$+left$(bl$,10-len(f1$))
310 rc$=rc$+f2$+left$(bl$, 5-len(f2$))
320 rc$=rc$+f3$+left$(bl$,10-len(f3$))
330 rc$=rc$+f4$+left$(bl$,15-len(f4$))
340 print#2,"p"+chr$(2)+chr$(rn)+chr$(0)+chr$(1)
350 print#1,rc$
360 print"Tovabbi bevitel (i/n)?"
370 getx$:if x$<>"i" and x$<>"n"then 370
380 ifx$="i" then 140
390 close 1:close 2:end
```

Ennek a programnak a működési elvét jobban megérthetjük a soronkénti elemzés alapján:

100	egy 15 hosszúságú üresjel-füzér definiálása
105	egy 41 karakter hosszú rekordokból álló relatív file megnyitása
110	a 15-ös parancscsatorna megnyitása
120	a relatív file előkészítéséhez pozicionálás az utolsó rekord első byte-jára
130	az utolsó rekord felszabadítása és az inicializálás kezdete
140	képernyőtörlés
150–260	a rekord kulcsszám és az 1–4. mezők bevitele, a mezők hosszának ellenőrzése
270–290	a beírt adatok egyszer helyesbíthetők
300–330	a rekord előkészítése
340	pozicionálás az adott kulcsú rekord első byte-jára
350	a rekord felírása lemezre
360–380	további bevitel
390	a program vége

Írjunk ki néhány rekordot ezzel a programmal a file-ba. Ne feledkezzünk meg a program tárolásáról, mert a továbbiakban is szükségünk lehet rá.
Most írjunk egy olyan programot, amely a felírt rekordokat visszaolvassa:

P.22

```
100 open1,8,2,"test.rel,l,"+chr$(41)
110 open2,8,15
115 printchr$(147)
120 input"Rekordszam: ";rn
130 print#2,"p"+chr$(2)+chr$(rn)+chr$(0)+chr$(1)
140 input#1,rc$
150 if asc(rc$)=255 then print"A rekord ures":goto250
160 printrc$
250 close1:close2
```

Ez a rutin egy meghatározott rekordot beolvas a file-ból. Ha a rekord üres, azaz nem írtunk bele eddig tényleges adatot, akkor az adatok helyén CHR$(255) van, hiszen ezt írtuk bele előkészítéskor.

Ha a rekordot kitöltöttük, a program kiírja annak tartalmát a képernyőre. A kiírásból láthatjuk, hogy az 1–4. mezők mindig azonos helyeken vannak. Ha a rekordot ismét fel akarjuk bontani mezőkre, a MID$ utasítást használhatjuk. Ahhoz például, hogy az 1. mezőt kivehessük a rekordból, a rekord megkeresése után az alábbi utasításokat próbáljuk ki parancs üzemmódban:

```
FL$=MID$(RC$,1,10)
PRINT F1$
```

Most az 1. mező az F1$ változóban van. A rekordnak ez az „ízekre szedése" a fenti rutinba beépíthető.

P.23

```
100 open1,8,2,"test.rel,l,"+chr$(41)
110 open2,8,15
115 printchr$(147)
120 input"Rekordszam: ";rn
130 print#2,"p"+chr$(2)+chr$(rn)+chr$(0)+chr$(1)
140 input#1,rc$
150 if asc(rc$)=255 then print"A rekord ures":goto250
160 printrc$
170 f1$=mid$(rc$, 1,10)
180 f2$=mid$(rc$,11, 5)
190 f3$=mid$(rc$,16,10)
200 f4$=mid$(rc$,26,15)
210 print"Mezo 1: ";f1$
220 print"Mezo 2: ";f2$
230 print"Mezo 3: ";f3$
240 print"Mezo 4: ";f4$
250 print"Tovabbi olvasas (i/n)?"
260 getx$:ifx$<>"i"andx$<>"n"then260
270 ifx$=""i"then115
280 close1:close2
```

Lényeges, hogy a MID$ utasításban lévő adatok a rekordon belül megfeleljenek a mező pontos pozíciójának. A zárójelek közötti első adat az a füzér, amelyet részeire szedünk. A második paraméter az a pozíció, ahol a füzéren belül a mező kezdődik, a harmadik pedig a mező hossza.

Eddig a rekordokat az INPUT utasítással olvastuk be. Ha azonban a rekord hossza meghaladja a 88 karaktert, az INPUT utasítás nem használható. Használjunk helyette GET utasítást,

amely a rekord byte-jait egyenként olvassa, s egy füzérré kapcsolja össze. Tegyük fel, hogy egy 128 karaktert tartalmazó relatív file-t készítettünk elő és töltöttünk fel adatokkal. A file 10. rekordját akarjuk az RC$ változóba beolvasni.

P.24

```
100 open 1,8,2,"test.get,l,"+chr$(128)
110 open 2,8,15
120 print#2,"p"+chr$(2)+chr$(10)+chr$(0)+chr$(1)
130 rc$=""
140 fori=1to128
150 get#1,x$
160 rc$=rc$+x$
170 nexti
180 printrc$
190 close1:close2
```

A fenti rutin lefutása után a rekord az RC$ változókban áll rendelkezésre. Ha ezt a rekordot egy PRINT utasítással írtuk fel pontosvessző nélkül, akkor az RC$ füzérben az utolsó karakter egy RETURN. A helyes megoldásban tehát a 140-es sorban lévő ciklust csak a 127-ig kell futtatni.

Mint már tudjuk, a P utasítás utolsó paramétere határozza meg, hogy az olvasás a rekord melyik karakterétől kezdődjön. Ha például az előző feladat 127 karakteres rekordjában a 40–60. pozíciójú mezőt szeretnénk olvasni, az utolsó paramétert a pozicionálásban 40-re kell állítani.

P.25

```
100 open 1,8,2,"test.get,l,"+chr$(128)
110 open 2,8,15
120 print#2,"p"+chr$(2)+chr$(10)+chr$(0)+chr$(40)
130 f$=""
140 fori=1to21
150 get#1,x$
160 f$=f$+x$
170 nexti
180 printf$
190 :
200 rem tovabbi ciklusok, feldolgozas
210 :
220 close1:close2
```

Miután a 120. sor a 10. rekord 40. byte-jára pozicionál és a 140–170-es sorokban a ciklus 21 byte-ot olvas be az F$-ba (a rekord 40–60. byte-jai), a ciklus lefutása után az ott lévő mező az F$-ban lesz.

A fentiekből következik, hogy ha csak a rekord egyik részére van szükségünk, nem kell az egész rekordot beolvasni. Ezt a pozicionálás negyedik paramétere teszi lehetővé.

1.5.6. A relatív file lezárása

A relatív file lezárásakor ugyanúgy járunk el, mint a soros file lezárásakor. Mivel azonban a relatív file kezeléséhez a pozicionálásnál szükségünk van a 15-ös parancscsatornára, az utóbbit is le kell zárni. A CLOSE és OPEN utasításokban természetesen a file-számoknak meg kell egyezni.

1.5.7. A bináris keresés

Normális körülmények között minden rekord elérése a rekord sorszámával történik. Előfordulhat azonban, hogy egy relatív címfile-ban Szántó urat keresünk, a megfelelő rekordsorszám azonban nem ismeretes. Egyik lehetőség az, hogy minden rekordot beolvasunk, s összehasonlítjuk a Szántó névvel stb. Ez a megoldás rendkívül nagy időráfordítást igényel. Ha a file rekordjai már rendezettek, akkor hatékonyabb keresési eljárást is alkalmazhatunk. Ilyen eljárás a *bináris keresés.*

A bináris keresés csak rendezett adathalmazon használható, ezért, ha egy új rekordot hozzáveszünk a rendezett adathalmazhoz, azt a megfelelő helyre kell beilleszteni.

A bináris keresést egy egyszerű példa alapján meg lehet érteni. Ha egy telefonkönyvben a név ismeretében keressük a telefonszámot, bizonyára nem sorosan járunk el. Felütjük a telefonkönyv közepét és megnézzük, hogy a keresett név első betűje megtalálható-e a felütött oldalon. Amennyiben a keresett név kezdőbetűje az abc-ben előbb van, a telefonkönyv első részét megfelezzük és így tovább.

Bináris keresésnél, ha egy adott lépésben nem találtuk meg a kívánt rekordot, a rendezettség alapján megállapítjuk, hogy az a pillanatnyi pozíciótól számítva előre vagy hátra lehet és a keresést az így kapott részhalmaz felénél folytatjuk.

Tekintsük az alábbi növekvő sorba rendezett számhalmazt:

KULCSSZÁM	TARTALOM
1	1985
2	1999
3	2005
4	2230
5	2465
6	2895
7	3490
8	3539
9	4123
10	5000
11	5210
12	6450
13	6500
14	6550
15	6999

A felsorolt 15 rekord közül a 3490 tartalmú rekordot keressük. Nem tudjuk, hogy hol van tárolva.

Először megállapítjuk, hogy a file hány rekordot tartalmaz. Ebben az esetben 15-öt. Ezt a számot 2-vel elosztjuk. A file közepe 8, vagyis a 3539 tartalmú rekordról van szó. Most megnézzük, hogy ez a rekord a 3490 értéket tartalmazza-e. Ha nem, a keresett érték kisebb, vagy nagyobb a megtalált tartalomnál, ebben az esetben 3539-nél. Az összehasonlítás eredménye szerint most éppen kisebb. Így a keresett rekord az aktuális kulcsnál kisebb kulcsú rekordok között van. Így jutunk el a 2230 tartalmú 4. rekordhoz. Az összehasonlítási eredmény arra utal, hogy a 3490 nagyobb a 4. rekordnál megtalált 2230 tartalomnál. A harmadik keresés a 4. és 8. kulcsok közötti középre irányul, vagyis a 2897 tartalmú 6. rekordra. Az összehasonlítási eredmény most kisebb értékre utal; ez azt jelenti, hogy a következő lépésben megtaláljuk a keresett értéket.

Bináris keresésnél a szükséges lépések számát az alábbi képlet alapján határozhatjuk meg:

S = INT(LOG(N)/LOG(2) + 1), ahol

S a lépések száma, N pedig a file által tartalmazott rekordok száma. Egy 1000 rekordos rendezett relatív file-nál például maximálisan 10 lépés szükséges egy tetszőleges rekord megkereséséhez!
Állítsunk elő egy 15 rekordot tartalmazó rendezett relatív file-t, amelyben bináris keresést végezhetünk:

P.26

```
100 open 1,8,2,"bin.rel,l,"+chr$(5)
110 fori=1to15
120 read rc$
130 print#1,rc$
140 nexti
150 data 1985,1999,2005,2230,2465,2897,3490
160 data 4123,5000,5210,6450,6500,6550,6999
```

Itt nincs szükség a pozicionálási utasításra, miután a file-t az első tételtől az utolsóig teleírjuk. A mutató a relatív file megnyitása után az első rekordon áll. Ebben a file-ban kell binárisan keresnünk a rekordokat:

P.27

```
100 open 1,8,2,"bin.rel,l,"+chr$(5)
110 open2,8,15
120 printchr$(147)
140 n=15: rem a rekordok szama
150 i=log(n)/log(2)
160 ifi-int(i)<>0theni=int(i)+1
170 m=i-1
180 i=2^i
190 x=i/2
210 input"Keresett ertek  (* - vege)";sb$
220 ifsb$="*"then320
230 ifm<0 then print"Nincs ilyen rekord":goto140
240 m=m-1
250 print#2,"p"+chr$(2)+chr$(x)+chr$(0)+chr$(1)
260 input#1,rc$
270 ifsb$=rc$then340
280 ifsb$<rc$then x=x-2^m:goto230
290 x=x+2^m
300 ifx>i then print"A file-t tulleptuk":goto140
310 goto230
320 close1:close2
330 end
340 print"A rekord tartalma: ";rc$
360 goto 140
```

A program dokumentálása:

100	a BIN.REL relatív file megnyitása
110	a parancscsatorna megnyitása
120	képernyőtörlés
140	a rekordok száma az N változóban van
150–190	Ha N nem hatványa 2-nek, képezzük 2-nek az N-nél nagyobb, de hozzá legközelebb eső hatványát. Ekkor a file-tartomány felfelé bővül, de egyetlen rekordot sem ha-

gyunk ki. A hatványkitevőt indexként használjuk. X az aktuális rekord sorszáma, ennek értéke a hatványérték fele, amely pontosan a file aktuális részhalmazának közepére mutat.

210–220 a keresett érték beolvasása. A program befejezése * karakter beütésével.

230 ha M<0, a keresett érték nincs az adathalmazban

240 M eggyel csökken. Az M-mel történő következő hatványra emelés tehát a file megmaradó részének a fele.

250–260 pozicionálás az X. sorszámú rekordra.

270 ha a beolvasott rekord megfelelt a keresett értéknek, a keresés megszakad, a rekord tartalmát kiírjuk.

280–310 megállapítjuk, hogy a keresett érték kisebb vagy nagyobb az olvasott rekord tartalmánál. Ennek megfelelően az X változó a felső vagy az alsó maradék középértékét tárolja.

320–330 A file-ok lezárása és a program befejezése

340–360 a megtalált rekord kiírása.

Ez egy általános célú binárisan kereső BASIC program. Csupán a rekordok számát, valamint a keresett értéket és a rekordok közötti összehasonlítást kell kicserélni benne, az aktuális feladatnak megfelelően.

1.5.8. Egy rekord keresése index-file segítségével

Ha gyakran van szükségünk bizonyos rekordokra, s ezeket olyan alfanumerikus kulcsokkal szeretnénk gyorsan elérni, amelyek nem felelnek meg a logikai rekord-sorszámnak, és a file-unkat nem akarjuk megfelelő rendezett formában tárolni, egy másik módszert kell a keresésre alkalmazni.

Minden kiválasztott adattípushoz ki kell alakítani egy saját index-file-t, amely rekordonként az alábbiakat tartalmazza:

- a mindenkori index
- a hozzátartozó kulcs.

Amikor szükség van rá, ezt a file-t betöltjük a tárba. Egy példa: a címkezelésünkhöz egy olyan relatív file-t állítottunk össze, amely a következőkből áll:

- vezetéknév
- keresztnév
- utca
- irányítószám
- lakhely
- telefonszám.

A keresést a keresztnév és a vezetéknév szerint egyaránt szeretnénk lehetővé tenni. Alakítsunk ki tehát még két relatív file-t, amelyek csak a kívánt mezőt – pl. a keresztnevet – és a fő-file-ban lévő megfelelő rekord kulcsát tartalmazzák.

A kívánt index-file-okat teljes egészükben mindig a tárban tartjuk, miután itt van lehetőség a leggyorsabb indexkeresésre. Ha például a TAMAS keresztnevű rekordot akarjuk elérni, menjünk végig a tárban a megfelelő index-file-on, majd a megtalált kulccsal közvetlenül olvassuk be a címfile kívánt rekordját.

Tegyük fel, hogy a vezetéknevekre nézve egy fő-file és egy index-file áll rendelkezésre:

Fő-file:			*Index-file:*		
VEZETÉKNÉV	KERESZTNÉV	TOVÁBBI MEZŐK	INDEX (vezetéknév)	KULCS LB	HB
Török	Tamás	....	Török	01	00
Máté	Csaba	....	Máté	02	00
Rendik	Zoltán	....	Rendik	03	00
Molnár	Lajos	....	Molnár	04	00
.	.		.		
.	.		.		
.	.		.		
Terpák	Csaba	....	Terpák	99	00

A file tehát 99 rekordot tartalmaz. Ahhoz, hogy a programot futtathassuk, be kell olvasni az indextáblázatot. Ez lehet egy olyan soros file, amelyet a DIM IT$ (99) utasítással deklarált változóba olvasunk be. Az IT$(99) tömbváltozó minden eleme tartalmaz egy vezetéknevet (első 20 karakter), továbbá egy alsó byte és egy felső byte értéket, amelyek az adott vezetéknevet tartalmazó rekord kulcsát határozzák meg. Az alábbi rutinnal egy tetszőleges rekord megkereshető.

```
100 INPUT "VEZETEKNEV";N$
110 FOR I=1T099
120 IF LEFT$(IT$,20)=N$THEN 150
130 NEXT I
140 PRINT "ILYEN NEV NINCS!":END
150 PRINT "MEGTALALTAM!"
160 OPEN 1,8,2,"CIMEK,L,"+CHR$(81)
170 OPEN 2,8,15
180 PRINT#2,"P"+CHR$(2)+MID$(IT$,21,1)+CHR$(0)+CHR$(1)
190 INPUT#1,RC$
         .
         .
         .
```

A 110–130. sorok által alkotott ciklus sorosan fut át az index táblázaton, és a keresett nevet összehasonlítja az első 20 karakterrel.
Amennyiben a vezetéknevet nem sikerült megtalálni, a rutin kilép a ciklusból s mielőtt a program befejeződne, a 140. sorban megjelenik a megfelelő üzenet.
Ha a rutin a 120. sorban megtalálja a keresett vezetéknevet, a 150. sorhoz ágazik le. Kiírja a „Megtaláltam" üzenetet, majd megnyitja a fő-file-t és a parancscsatornát. Pozicionál a megfelelő kulcsú rekordra. (Az alsó és felső byte-ok értékét MID$ függvénnyel az aktuális indexelemből állítja elő.)
Ezután a 190. sorban beolvassa a rekordot.
Az index-file-okkal történő keresés szintén gyors és rendkívül rugalmas formája az adatszervezésnek. Elméletileg minden fő-file-hoz tetszőleges számú index-file-t alkalmazhatunk, szem előtt tartva az alábbiakat:

1. A fő-file-ban a mezőket érintő változtatásoknál a megfelelő index-file-okon is végre kell hajtani a módosítást. Ez, különösen több index-file esetén rendkívül munkaigényes lehet.
2. Minthogy az index-file-okat, a gyors elérés érdekében a tárban kell tartanunk, azok számát a tárkapacitás korlátozza.

1.5.11. HÁZTARTÁSI NAPLÓ relatív adattárolással

Ebben a fejezetben egy kisebb adatfeldolgozási rendszer programját mutatjuk be, amit bárki felhasználhat a saját háztartásának nyilvántartására, vagy alapul szolgálhat egy bonyolultabb rendszer kifejlesztéséhez.
Az alapadatokat relatív file-ban fogjuk tárolni, tehát a rendszer kulcsos adatszerkezetre épül. Tegyük fel, hogy valaki nyilván akarja tartani háztartási kiadásait, azzal a céllal, hogy mindig napra készen tudja összesíteni és visszamenőleg rendszerezni a különböző jellegű kiadásokat. A feladat megoldásához az első lépés a tárolni kívánt adatok szerkezetének megállapítása. A nyilvántartás alapját a számlatípusok képezik, a tárolt adatállomány egy rekordja tehát egy számlatípus lesz. Ha a számlatípusokat sorrendben megszámozzuk, ez a sorszám egyedi, tehát képezheti a relatív file kulcsát. Tegyük fel, hogy háztartásunkban 20 féle számlatípus fordul elő, a relatív file tehát 20 rekordot fog tartalmazni.
Milyen legyen a rekord belső szerkezete?
Ha a számlatípusokat csak a kódokkal különböztetjük meg egymástól, a feldolgozás során mindig fejből kell tudnunk, hogy melyik kódnak milyen számlatípust feleltettünk meg az adathalmaz létrehozásakor. Saját munkánk megkönnyítésére célszerű tehát a számlatípus szöveges megnevezését is tárolni.
Legyen minden számlarekord első mezője egy 20 karakteres szöveges megnevezés.
Kiadásainkat és bevételeinket havi bontásban célszerű nyilvántartani, ezért minden rekordban 12, egyenként 10 karakteres mezőben tároljuk a havi göngyölített összegeket. Az, hogy a tárolás karakteres változókkal történik, nem zavaró, hiszen a VAL függvénnyel ezeket a változókat bármikor konvertálhatjuk, ha számításokat akarunk végezni velük.
A rekord hossza 20+12*10+RETURN=141 karakter, felépítése pedig a következő:

MEZŐ	HOSSZÚSÁG	POZÍCIÓ
A számla neve	20	1–20
Januári összeg	10	21–30
Februári összeg	10	31–40
.		
.		
Novemberi összeg	10	121–130
Decemberi összeg	10	131–140

Kiadásainkat visszamenőleg több évre tárolhatjuk, ha minden év adatait külön file-ba helyezzük el. Egy év adatait egy 20 rekordból álló, egyenként 141 karaktert tartalmazó relatív file tartalmazza.
Minden adatfeldolgozási feladat megoldásának kiindulópontját a fenti elemzések alkotják, azzal az eltéréssel, hogy nem minden adathalmaz szerkezete ilyen egyszerű. A feladat megoldásának ezt az előkészítő fázisát *rendszerszervezésnek* nevezzük.
A rendszerszervezés második lépése azoknak a funkcióknak a meghatározása, amelyeket az adathalmazon dolgozó programnak el kell tudni végezni.
A mi nyilvántartási rendszerünket az alábbi feladatok ellátására szeretnénk felkészíteni:
- számlák előkészítése
- könyvelés
- számlaáttekintés

```
380 fori=1to4
390 print"Mezo";i;": ";f$(i)
400 nexti
410 print"------------------------------------"
420 rem ==============================
430 rem     mezo modositasa
440 rem ==============================
450 print"Valassza ki a modositando mezo szamat!"
460 getx$:if x$<"1" or x$>"4" then 460
470 input"Uj tartalom: ";f$(val(x$))
480 print"Ebben a rekordban modosit meg (i/n)?"
500 getx$:if x$<>"i" and x$<>"n" then 500
510 if x$="i"then340
520 rem ==============================
530 rem     mezok osszerakasa
540 rem ==============================
545 rc$=""
550 rc$=f$(1)+left$(bl$,10-len(f$(1)))
560 rc$=rc$+f$(2)+left$(bl$, 5-len(f$(2)))
570 rc$=rc$+f$(3)+left$(bl$,10-len(f$(3)))
580 rc$=rc$+f$(4)+left$(bl$,15-len(f$(4)))
590 rem ==============================
600 rem     rekord visszairas
610 rem ==============================
620 print#1,rc$
630 rem ==============================
640 rem     program befejezes?
650 rem ==============================
660 print"Tovabbi adatmodositas (i/n)?"
670 getx$:if x$<>"i" and x$<>"n" then 670
680 if x$="i"then160
690 close1:close2
```

A módosított program segítségével bármely rekordot és azon belül bármely mezőt megváltoztathatunk, de ügyeljünk arra, hogy a program az új mezők hosszát nem ellenőrzi, felesleges hosszukat egyszerűen levágja.

1.5.10. A relatív file bővítése

A relatív file-ban tárolt rekordok számát a felhasználók általában előre megbecsülik. Célszerű ezzel a becsült értékkel lefuttatni az 1.5.4. alfejezetben ismertetett előkészítő programot. Ez azonban nem jelenti azt, hogy a megbecsült rekordszámot a feldolgozás során nem léphetjük túl.

Ha a lemez kapacitása megengedi, a létrehozott relatív file bármikor bővíthető. Célszerű az újonnan meghatározott legnagyobb sorszámú rekordba CHR$(255)-öt írni, ekkor ugyanis a korábbi és az új file végződése közötti összes rekordok felszabadulnak.

Az elmondottak alapján tehát nem okoz gondot egy olyan rekord felírása a relatív file-ba, amely túllépi a file végét. Igaz, hogy ez a lépés a RECORD NOT PRESENT hibaüzenetet eredményezi, de a rekord felírását a DOS elvégzi. Ha viszont a file bővítésére kevés a lemez szabad területe, a FILE TOO LARGE hibaüzenetet olvashatjuk le a hibacsatornáról.

A RECORD NOT PRESENT hibaüzenet olvasás közben tényleges hibát jelez, nevezetesen, hogy egy nem létező rekordot próbálunk beolvasni.

1.5.11. HÁZTARTÁSI NAPLÓ relatív adattárolással

Ebben a fejezetben egy kisebb adatfeldolgozási rendszer programját mutatjuk be, amit bárki felhasználhat a saját háztartásának nyilvántartására, vagy alapul szolgálhat egy bonyolultabb rendszer kifejlesztéséhez.
Az alapadatokat relatív file-ban fogjuk tárolni, tehát a rendszer kulcsos adatszerkezetre épül. Tegyük fel, hogy valaki nyilván akarja tartani háztartási kiadásait, azzal a céllal, hogy mindig napra készen tudja összesíteni és visszamenőleg rendszerezni a különböző jellegű kiadásokat. A feladat megoldásához az első lépés a tárolni kívánt adatok szerkezetének megállapítása. A nyilvántartás alapját a számlatípusok képezik, a tárolt adatállomány egy rekordja tehát egy számlatípus lesz. Ha a számlatípusokat sorrendben megszámozzuk, ez a sorszám egyedi, tehát képezheti a relatív file kulcsát. Tegyük fel, hogy háztartásunkban 20 féle számlatípus fordul elő, a relatív file tehát 20 rekordot fog tartalmazni.
Milyen legyen a rekord belső szerkezete?
Ha a számlatípusokat csak a kódokkal különböztetjük meg egymástól, a feldolgozás során mindig fejből kell tudnunk, hogy melyik kódnak milyen számlatípust feleltettünk meg az adathalmaz létrehozásakor. Saját munkánk megkönnyítésére célszerű tehát a számlatípus szöveges megnevezését is tárolni.
Legyen minden számlarekord első mezője egy 20 karakteres szöveges megnevezés.
Kiadásainkat és bevételeinket havi bontásban célszerű nyilvántartani, ezért minden rekordban 12, egyenként 10 karakteres mezőben tároljuk a havi göngyölített összegeket. Az, hogy a tárolás karakteres változókkal történik, nem zavaró, hiszen a VAL függvénnyel ezeket a változókat bármikor konvertálhatjuk, ha számításokat akarunk végezni velük.
A rekord hossza 20 + 12•10 + RETURN = 141 karakter, felépítése pedig a következő:

MEZŐ	HOSSZÚSÁG	POZÍCIÓ
A számla neve	20	1–20
Januári összeg	10	21–30
Februári összeg	10	31–40
.		
.		
Novemberi összeg	10	121–130
Decemberi összeg	10	131–140

Kiadásainkat visszamenőleg több évre tárolhatjuk, ha minden év adatait külön file-ba helyezzük el. Egy év adatait egy 20 rekordból álló, egyenként 141 karaktert tartalmazó relatív file tartalmazza.
Minden adatfeldolgozási feladat megoldásának kiindulópontját a fenti elemzések alkotják, azzal az eltéréssel, hogy nem minden adathalmaz szerkezete ilyen egyszerű. A feladat megoldásának ezt az előkészítő fázisát *rendszerszervezésnek* nevezzük.
A rendszerszervezés második lépése azoknak a funkcióknak a meghatározása, amelyeket az adathalmazon dolgozó programnak el kell tudni végezni.
A mi nyilvántartási rendszerünket az alábbi feladatok ellátására szeretnénk felkészíteni:
- számlák előkészítése
- könyvelés
- számlaáttekintés

– számlanevek kiírása
– havi áttekintás
– éves áttekintés

A számlák előkészítése

Ez a rutin összeállítja az évi file-t. Lekérdezi a létező számlatípusok számát és megnevezését. Felépíti az egyes számlatípusok rekordjait, az összegmezőkbe nullát ír. Ha már létezett ez a file, ezzel a rutinnal törölhetjük és újra felépíthetjük.

A könyvelés

A könyvelendő számla számának beírása után meg kell adni, hogy a számlán levő összeg bevétel vagy kiadás.
A gép ekkor kiírja az aktuális számlaértéket, bekönyveljük a megfelelő összeget, amely mindig pozitív. Ha korrekciós könyvelést végzünk (azaz javítunk), negatív összeget is megadhatunk. Végül a rutin kiírja az új számlaértéket és átléphetünk a következő tétel könyvelésére.

A számlaáttekintés

A számlaszám beírása után megkapjuk az egyes hónapokban elkönyvelt összegeket és az évi teljes összeget.

A számlanevek kiíratása

Az adathalmaz létrehozásakor minden számlatípust számlaszámmal azonosítottunk. Előfordulhat azonban, hogy valamelyik számlatípus azonosítóját elfelejtettük. Ez a rutin kilistázza az azonosítókat a hozzájuk tartozó megnevezésekkel együtt.

Havi áttekintés

Ha arra vagyunk kíváncsiak, hogy egy adott hónapban az egyes számlatípusokon milyen pénzforgalmat bonyolítottunk le, ezzel a rutinnal lekérdezhetjük. A rutin számlatípusonként kilistázza egy adott hónap pénzforgalmát.

Éves áttekintés

Az egész évi pénzforgalom áttekintéséhez (számítógép nélkül) át kellene lapoznunk a teljes számlakészletünket. Végül is ebben a rutinban a gép sem tesz mást, mint végigolvassa az összes számlatípus rekordját ahhoz, hogy havi és évi egyenleget készíthessen. A többi rutinhoz képest ennek a programrésznek a futtatása a legidőigényesebb.

Ezzel gyakorlatilag befejeztük a háztartási könyvelési program áttekintését. Alapos tanulmányozásával azonban sok ötletet meríthetünk bonyolultabb adatfeldolgozó rendszerek elkészítéséhez.
A program listája:

P.29

```
100 poke53280,2:poke53281,2:printchr$(158);:bl$="                        ":dims(12)
110 gosub2050
120 input"aktualis ev : ";j$
130 ifj$<"1984"ory$>"1999"thenprintchr$(145);:goto120
140 gosub2050
150 print"Valasszon !"
160 print"--------------------":print
170 print"      -1- Szamlak elokeszitese"
180 print"      -2- Konyveles"
```

```
190 print"      -3- Szamlaattekintes"
200 print"      -4- Szamlanevek kiirasa"
210 print"      -5- Havi attekintes"
220 print"      -6- Eves attekintes":print
230 print"      -0- Program befejezese"
240 get x$:if x$<"0" or x$>"6" then 240
250 if x$<>"0"then270
260 end
270 on val(x$)gosub 290,560,920,1160,1370,1720
280 goto140
290 rem ====================
300 rem  szamla felfektetes
310 rem ====================
320 gosub2050
330 print"Figyelem! Valamennyi ezevi adatfile      "
340 print"torlesre kerul!":print
350 print"folytassam (i/n)?"
360 getx$:ifx$<>"i"and x$<>"n"then 360
370 if x$="i"then390
380 close1:close2:return
390 open2,8,15,"s:szamlak"+j$
400 open1,8,2,"szamlak"+j$+",l,"+chr$(141)
410 gosub2050
420 input"A szamlak szama  (1-20)? : ";kz
430 print
440 ifkz<1orkz>20thenprintchr$(145);:goto420
450 fori=1tokz
460 print"A szamla  neve:    ";i;": ";
470 inputkn$
480 iflen(kn$)>20thenprintchr$(145);:goto460
490 rc$=kn$+left$(bl$,20-len(kn$))
500 forx=1to12
510 rc$=rc$+str$(0)+left$(bl$,8)
520 nextx
530 print#1,rc$
540 next i
550 close1:close2:return
560 rem ===============
570 rem   konyveles
580 rem ===============
590 gosub2050
600 input"A szamla szama ";kn
610 ifkn<1orkn>20thenprintchr$(145);:goto600
620 gosub2140
630 print"-----------------------------------"
640 printkn;" - ";kn$
650 print"-----------------------------------"
660 print"bevetel vagy kiadas (b/k)?"
670 print"-----------------------------------"
680 getx$:ifx$<>"b"andx$<>"k"then680
690 input"honap (1-12)     : ";m
700 ifm<1orm>12thenprintchr$(145);:goto690
710 print"-----------------------------------"
720 print"elozo szamlaertek : ";s(m)
730 print"-----------------------------------"
740 input"konyvelendo osszeg:";bb
750 print"-----------------------------------"
760 ifx$="b"then s(m)=s(m)+bb:goto780
770 s(m)=s(m)-bb
780 print"jelenlegi osszeg: ";s(m)
790 print"-----------------------------------"
800 rc$=kn$+left$(bl$,20-len(kn$))
810 fori=1to12
820 s$=str$(s(i))
830 rc$=rc$+s$+left$(bl$,10-len(s$))
840 nexti
850 print#2,"p"+chr$(2)+chr$(kn)+chr$(0)+chr$(1)
```

```
860 print#1,rc$
870 close1:close2
880 print"Tovabbi konyveles (i/n)?"
890 getx$:ifx$<>"i"andx$<>"n"then890
900 ifx$="i"thengosub2050:goto600
910 return
920 rem ==================
930 rem  szamlaattekintes
940 rem ==================
950 gosub2050
960 input"A szamla szama : ";kn
970 ifkn<1orkn>20thenprintchr$(145);:goto960
980 gosub2140
990 gosub2050:printchr$(145);chr$(145);
1000 print"------------------------------"
1010 printkn;" - ";kn$
1020 print"------------------------------"
1030 print"Havi egyenleg"
1040 print"------------------------------"
1050 gs=0
1060 fori=1to12
1070 printi;tab(8);s(i)
1080 gs=gs+s(i)
1090 nexti
1100 print"------------------------------"
1110 print"Osszes:";tab(8);gs
1120 printtab(8);"======"
1130 print"tovabb - [RETURN]"
1140 inputx$
1150 close1:close2:return
1160 rem ======================
1170 rem  szamlanevek kiirasa
1180 rem ======================
1190 gosub2050
1200 open1,8,2,"szamlak"+j$+",l,"+chr$(141)
1210 open2,8,15
1220 fori=1to20
1230 print#2,"p"+chr$(2)+chr$(i)+chr$(0)+chr$(1)
1240 rc$=""
1250 forx=1to20
1260 get#1,x$
1270 rc$=rc$+x$
1280 nextx
1290 input#2,x
1300 ifx=50then 1340
1310 gs=gs+val(s$)
1320 printi;"  -  ";rc$
1330 i=i+1:goto1230
1340 print"tovabb - [RETURN]"
1350 inputx$
1360 close1:close2:return
1370 rem ---------------
1380 rem havi attekintes
1390 rem ---------------
1400 gosub2050
1410 input"honap : ";m
1420 gosub2050
1430 print"---------------------------------------"
1440 print"sz.  sz.nev                 osszeg  "
1450 print"---------------------------------------"
1460 open1,8,2,"szamlak"+j$+",l,"+chr$(141)
1470 open2,8,15
1480 gs=0
1490 for kn=1to20
1500 kn$="":s$=""
1510 print#2,"p"+chr$(2)+chr$(kn)+chr$(0)+chr$(1)
1520 for i=1to20
```

```
1530 get#1,x$
1540 kn$=kn$+x$
1550 nexti
1560 input#2,f
1570 iff<>50then 1590
1580 goto1670
1590 print#2,"p"+chr$(2)+chr$(kn)+chr$(0)+chr$(20+(m-1)*10)
1600 for i=1to10
1610 get#1,x$
1620 s$=s$+x$
1630 next i
1640 gs=gs+val(s$)
1650 print kn;tab(6);kn$;tab(26);s$
1660 next kn
1670 print"------------------------------------"
1680 print"Egyenleg      ";tab(26);str$(gs)
1690 printtab(26);"======="
1700 print"tovabb - [RETURN]"
1710 inputx$:close1:close2:return
1720 rem ================
1730 rem eves attekintes
1740 rem ================
1750 gosub2050
1760 open1,8,2,"szamlak"+j$+",l,"+chr$(141)
1770 open2,8,15
1780 print"----------------------------------------"
1790 print"sz.  szamla nev            eves egyenl."
1800 print"----------------------------------------"
1810 gs=0
1820 for kn=1to20
1830 print#2,"p"+chr$(2)+chr$(kn)+chr$(0)+chr$(1)
1840 rc$=""
1850 fori=1to140
1860 get#1,x$
1870 rc$=rc$+x$
1880 nexti
1890 input#2,f:iff=50then1980
1900 kn$=left$(rc$,20)
1910 js=0
1920 fori=1to12
1930 js=js+val(mid$(rc$,20+(i-1)*10,10))
1940 nexti
1950 gs=gs+js
1960 printkn;tab(6);kn$;tab(26);js
1970 nextkn
1980 print"--------------------------------------"
1990 close1:close2
2000 print"Egyenleg      ";tab(26);gs
2010 printtab(26);"======="
2020 print"tovabb - [RETURN]"
2030 inputx$
2040 return
2050 rem ====================
2060 rem    program fejresz
2070 rem ====================
2080 printchr$(147);
2090 printtab(4);"=================================="
2100 printtab(4);"  H A Z I  szamla-konyveles  "+j$
2110 printtab(4);"=================================="
2120 print:print
2130 return
2140 rem ================
2150 rem szamlak beolvasasa
2160 rem ================
2170 open1,8,2,"szamlak"+j$+",l,"+chr$(141)
2180 open2,8,15
2190 print#2,"p"+chr$(2)+chr$(kn)+chr$(0)+chr$(1)
```

```
2200 rc$=""
2210 fori=1to140
2220 get#1,x$
2230 rc$=rc$+x$
2240 next i
2250 input#2,f
2260 iff<>50then 2300
2270 print"Eves file, vagy szamla nincs!!!":print
2280 print"tovabb - [RETURN]"
2290 close1:close2:return
2300 kn$=left$(rc$,20)
2310 gs=0
2320 fori=1to12
2330 s(i)=val(mid$(rc$,20+(i-1)*10,10))
2340 gs=gs+s(i)
2350 next i
2360 return
2370 open15,8,15
2380 input#15,x1$,x2$
2390 printx1$,x2$
2400 close15
```

A program dokumentálása:

A program indításához szükséges előkészítések:

100	Beállítjuk a képernyő és a karakterek színét, feltöltünk egy változót üres karakterekkel, deklaráljuk a számlaösszeg-változót.
110–130	Kiírjuk a fejlécet és bekérjük az aktuális évet.
140–280	Kiírjuk a menüt és beolvassuk a felhasználó választását, majd meghívjuk a kívánt rutint.

A számlák előkészítése:

390–400	Az esetlegesen létező évi file-t töröljük, az új file-t megnyitjuk.
480	Az RC$ rekord 1–20. pozícióira elhelyezzük a felhasználó által beírt megnevezést.
500–540	A havi összegeket kinullázzuk, beillesztjük a rekordba.
530	A rekordot beírjuk a file-ba.

Könyvelés:

590	Meghívjuk a „számla beolvasása" rutint. Ez a rutin a számla havi összegeit S(1)-től S(12)-ig változókba helyezi.
660	Megkérdezzük, hogy a könyvelendő összeg bevétel vagy kiadás.
800	A számlanevet elhelyezzük a rekordba.
810–840	A számlaösszeget beírjuk a rekordba.
850–860	A módosított rekordot visszaírjuk a file-ba.

A számla áttekintése:

980	A kívánt számlát beolvassuk és a havi összegeket előkészítjük az S(1)-től S(12)-ig terjedő változókban.
1050–1090	A havi összegeket kiírjuk és göngyölítjük a GS változóban.
1110	A teljes összeget kiírjuk.

A számlanevek kiírása:

1220	Beállítjuk a kezdő számlaszámot.
1230	Pozicionálás a megfelelő számla rekordjára.
1240–1280	A számla megnevezését beolvassuk az RC$ változóba.

1290–1300 Ha az 50 RECORD NOT PRESENT hiba lép fel a csatornán, visszatérünk a főprogramunkba.
1320 Kiírjuk a számlaszámot és a megnevezést.

Havi áttekintés:
1490–1660 Beolvassuk az összes számlát.
1510 Pozicionálunk a rekordra.
1520–1550 Beolvassuk a számla nevét.
1560–1580 Megvizsgáljuk, hogy a számla valóban létezik-e.
1590 Pozicionálunk a kívánt hónap összegmezőjére.
1600–1630 Beolvassuk a havi összeget.
1640 A havi összegeket összeadjuk.
1650 A számlaszám, a számlanév és a havi összeg kiírása.
1680 Kiírjuk az egyenleget.

Éves áttekintés:
1820–1970 Beolvassuk az összes számlát.
1830 Pozicionálunk a rekordra.
1850–1880 Beolvassuk a rekordot a változóba.
1890 Ellenőrizzük, hogy fellépett-e a RECORD NOT PRESENT hiba.
1900 Meghatározzuk a számla megnevezését.
1920–1940 Beolvassuk a havi összegeket, numerikus alakra konvertáljuk és gyűjtjük a JS változóba.
1950 Kiszámítjuk a végösszeget (GS).
1960 Kiírjuk a számlaszámot, a számla nevét és az évi összeget.
2000 Kiírjuk az egyenleget.

A számlák beolvasása:
2190 Pozicionálás a KN kulcsú rekordra.
2210–2240 A rekordot beolvassuk az RC$-ba.
2250–2260 Ellenőrizzük, hogy fellépett-e a RECORD NOT PRESENT hiba.
2300 A rekordból kiolvassuk a számlaszámot.
2320–2350 A havi összegeket kiolvassuk a rekordból, átalakítjuk numerikus formába és az S(1)-től S(12)-ig terjedő változókat feltöltjük.

1.6. A lemezegység hibaüzenetei és ezek okai

Ha lemezkezeléskor hibát vétünk vagy fizikai lemezhiba lép fel, a vörös LED villogni kezd az egységen. A LED mindaddig villog, amíg a hibaüzenetet le nem olvassuk, vagy amíg nem küldünk egy új utasítást a lemezegységhez.

Mindenekelőtt azt nézzük meg, hogyan olvasható le a lemezegység hibaüzenete.
Meg kell nyitnunk a 15 másodlagos című hiba- vagy parancscsatornát:

```
100 OPEN 15,8,15
110 INPUT#15, A,B$,C,D
120 PRINT A,B$,C,D
```

Ha nem volt hiba, a rutin a következő üzenetet írja ki:

```
Ø   OK   Ø   Ø
```

Itt az első szám (A) a hiba száma, a mi esetünkben tehát 0: nincs hiba. Ezután következik a szöveges hibaüzenet (B$ változó). A C és a D változó azt a sáv- és szektorszámot tartalmazza, ahol a hiba fellépett, ha ezt a hiba jellege megengedi (főként hardverhibáknál és blokk-orientált utasításoknál).
A hibacsatornát a következőképpen is leolvashatjuk:

```
100 OPEN 15,8,15
110 GET#15,A$:PRINT A$;:IF ST<>64 THEN 110
ØØ, OK,ØØ,ØØ
```

A következőkben ismertetjük az összes lehetséges hibaüzenetet:

00, OK, 00, 00
Ez az üzenet abban az esetben jelentkezik, ha az utolsó lemezművelet hibamentesen zajlott le, vagy az utolsó hibaüzenet olvasása után nem érkezett adat vagy utasítás az egységhez.

01, FILES SCRATCHED, XX, 00
Ez egy SCRATCH utasítást követő visszajelzés. Az XX szám adja meg a törölt file-ok számát, hiszen több file-t is törölhetünk a JOKER utasítás használatakor. Ez tulajdonképpen nem hibaüzenet, így a LED sem villog. Ha a BASIC 4.0 SCRATCH utasítással dolgozunk, a rendszer a visszajelzést automatikusan behozza és kijelzi.

20, READ ERROR, TT, SS
Olvasási hiba, amely azt jelenti, hogy nem sikerült megtalálni a *blokk fejlécében* (header). Ez a hiba általában hibás lemez olvasásakor lép fel. A TT és az SS arra a sávra és szektorra vonatkozik, amelynél a hiba fellépett. Teendő: a meghibásodott lemezt le kell cserélni.

21, READ ERROR, TT, SS
Ez is olvasási hiba. Egy blokkhoz nem sikerült megtalálni a megfelelő SYNC karaktert (szinkronkaraktert).
A hibaüzenet oka az lehet, hogy nincs az egységben lemez, vagy ha mégis van, nincs megformálva, vagy elállítódott az író/olvasófej. Teendő: megpróbáljuk formálni a lemezt, ha nem sikerül, kicseréljük. Ha az új lemezzel is baj van, be kell állítani az egység író/olvasófejét.

22, READ ERROR, TT, SS
Ez a hibaüzenet azt jelenti, hogy egy adatblokk fejlécében *kontrollsumma* (ellenőrzőösszeg) hiba van, ami egy blokk hibás írásából eredhet.

23, READ ERROR, TT, SS
Valószínű, hogy egy olyan adatblokkot olvastunk a DOS-pufferbe, amelyben kontrollsumma hiba van. Egy vagy több adatbyte hibás. Teendő: mentsük a lemezen levő file-okat másik lemezre, ha még lehetséges.

24, READ ERROR, TT, SS
Ennél a hibaüzenetnél is kontrollsumma hiba lépett fel az adatblokkban, vagy az előző blokk-fejlécben. Hibás file-okat próbálunk beolvasni. Teendő: ugyanaz, mint a 23. hibánál.

25, WRITE ERROR, TT, SS
Minden adatblokk felírása után az adatokat a DOS még egyszer visszaolvassa és összehasonlítja a pufferben lévő adatokkal (VERIFY). Ezt a hibát a rendszer abban az esetben jelzi, ha az összehasonlítás során nincs azonosság. Teendő: a hibát kiváltó utasítást meg kell ismételni. Ha ez az intézkedés eredménytelennek bizonyul, a megfelelő adatblokkot le kell zárni Block-Allocate utasítással (l. 2.2.4. fejezet).

26, WRITE PROTECT ON, TT, SS
Írásvédelem. A vezérlőnek egy blokkot kellene írnia a lemezre, holott írásvédelmet érzékelt, azaz a lemezen rajta van az írásvédelmi tapasz. Teendő: az írásvédelem eltávolítása.

27, READ ERROR, TT, SS
Olvasási hiba a fejlécben (headerben). A vezérlő hibát talál az adatblokk fejlécében. A DOS a blokkot ekkor nem olvassa be a tárba. Ez az üzenet földelési problémát is jelezhet. Teendő: az utasítás megismétlése vagy a blokk lezárása.

28, WRITE ERROR, TT, SS
Íráshiba (hosszú adatblokk). A vezérlő megpróbálja érzékelni a következő blokk szinkronjelét az adatblokk írása után. Ha a szinkronjel egy megadott időn belül nem jelenik meg, hibaüzenetet kapunk. Vagy hardver eredetű hiba lépett fel, vagy rossz a lemez formálása. Teendő: a lemez újraformálása vagy kicserélése.

29, DISK IN MISMATCH, TT, SS
Lemez ID-hiba. A hiba akkor jelentkezik, ha az egységben lévő lemez ID-je nem azonos a DOS-ban tárolt ID-vel. A hiba oka vagy egy blokk-fejléc megsérülése, vagy az, hogy nem inicializáltuk a lemezt. Teendő: a lemez inicializálása.

30, SYNTAX ERROR, 00, 00
Szintaktikus hiba. A DOS nem tudja értelmezni a parancscsatornára kapott utasítást. Teendő: az utasítás ellenőrzése és helyesbítése.

31, SYNTAX ERROR, 00, 00
Szintaktikus hiba (nem létező utasítás). A DOS olyan utasítást kapott, amely nem szerepel az utasításkészletében. Ilyen pl. a BACKUP DUPLICATE utasítás. Teendő: az utasítást csak BASIC bővítő programmal együtt használjuk.

32, SYNTAX ERROR, 00, 00
Szintaktikus hiba (hosszú sor). A kapott utasítás 40 karakternél hosszabb. Teendő: az utasítás lerövidítése.

33, SYNTAX ERROR, 00, 00
Szintaktikus hiba (nem létező file-név). Az OPEN vagy SAVE utasításban szabálytalanul használtuk a JOKER-t ("*", "?"). Teendő: a szöveget ki kell javítani.

34, SYNTAX ERROR, 00, 00
Szintaktikus hiba (nincs megadva a file). A DOS nem találja meg a file-nevet egy utasításban, mert pl. az utasításból lemaradt a ":". Teendő: az utasítás ellenőrzése.

39, FILE NOT FOUND, 00, 00
Szintaktikus hiba (nem létező file). Automatikus végrehajtáshoz nem sikerült megtalálni egy USR típusú felhasználói programot. Teendő: a file-név ellenőrzése.

50 RECORD NOT PRESENT, 00, 00
A rekord nincs jelen. Egy relatív file-nál olyan rekordot akarunk olvasni vagy írni, amelybe eddig még nem írtunk adatot. Írásnál ez tulajdonképpen nem hiba, az üzenet csak arra utal, hogy üres rekordra írunk. Írásnál a hibaüzenet elkerülhető, ha a relatív file létrehozásakor a legmagasabb kulcsú rekordra CHR$ (255) karaktert írunk.

51, OVERFLOW IN RECORD, 00, 00
Túlcsordulás a rekordban. A rekord hossza (beleértve a RETURN karaktert is) a relatív file-ba történő íráskor meghaladja az OPEN-ben megadott értéket. A fölösleges karakterek figyelmen kívül maradnak.

52, FILE TOO LARGE, 00, 00
A file túl nagy. A rekordok száma a relatív file-ban túl sok, meghaladja a lemezkapacitást. Teendő: másik lemez használata vagy a rekordok számának csökkentése.

60, WRITE FILE OPEN, 00, 00
A file írásra van nyitva. Ez az üzenet akkor keletkezik, ha olyan file-t akarunk olvasásra megnyitni, amit írásnál nem zártunk le. Teendő: az OPEN utasításban az M üzemmód használata file-olvasáshoz.

61, FILE NOT OPEN, 00, 00
A file nincs megnyitva. Ez az üzenet akkor keletkezik, ha a DOS-ban egy nem megnyitott file-hoz akarunk hozzáférni. Teendő: a file megnyitása, vagy a file-név ellenőrzése.

62, FILE NOT FOUND, 00, 00
Nincs meg a file. Egy olyan program betöltésével vagy egy olyan file megnyitásával próbálkoztunk, amely a lemezen nem létezik. Teendő: a file-név ellenőrzése.

63, FILE EXISTS, 00, 00
A file létezik. A létrehozott file neve a lemezen egyszer már megtalálható. Teendő: másik file-név vagy az @ használata.

64, FILE TYPE MISMATCH, 00, 00
File-típus hiba. Egy file megnyitásakor a file-típus nem egyezik a tartalomjegyzékben lévő file-típussal. Teendő: a file-típus helyesbítése.

65, NO BLOCK, TT, SS,
Nincs blokk. Ez a hibaüzenet a Block-Allocate utasításnál jelentkezik, ha a lefoglalandó blokk már nem szabad. A DOS ebben az esetben automatikusan egy, az első szabad blokkhoz tartozó sáv/szektor értékét írja ki. Ha már nincs szabad blokk, két 0-t ír ki.

66, ILLEGAL TRACK OR SECTOR, TT, SS
Illegális sáv vagy szektor. B-R utasítással olyan szektorhoz próbálunk hozzáférni, amely nem létezik.

67, ILLEGAL TRACK OR SECTOR, TT, SS
Egy file sáv/szektor láncolása egy nem létező sávra vagy szektorra mutat. (l. 3.4.3. alfejezet)

70, NO CHANNEL, 00, 00
Nincs csatorna. Több file-t próbáltunk megnyitni, mint amennyi csatorna rendelkezésre áll, vagy az adott közvetlen elérésű csatorna már foglalt.

71, DIR ERROR, TT, SS
Tartalomjegyzék hiba. A DOS-tárban levő szabad blokkok száma nem egyezik a BAM nyilvántartásával. Előfordulhat, hogy elmaradt a lemez inicializálása.

72, DISK FULL, 00, 00
A lemezen már 3-nál kevesebb szabad blokk van, vagy a rendszer elérte a tartalomjegyzék bejegyzések maximális számát (A VC 1541-nél: 144).

73, CBM DOS V2.6 1541, 00, 00
Ez az üzenet a VC 1541-es készülék bekapcsolási üzenete. Abban az esetben jelent hibaüzenetet, ha az írást egy olyan lemezen kíséreljük meg, amelynek formálása nem azonos DOS változattal történt, például a CBM 4040-nel, a CBM 3040 elődjével(1.0 DOS változat).

74, DRIVE NOT READY, 00, 00
Ez a hibaüzenet akkor jelentkezik, ha lemezműveletet akarunk végezni anélkül, hogy lemez lenne a meghajtóegységben vagy az egység be lenne kapcsolva.

75, FORMAT SPEED ERROR, 00, 00
A formálás során rossz a fordulatszám. Ezt a hibaüzenetet csak a CBM 8250 tartalmazza.

1.7. BASIC utasítások összehasonlító áttekintése

BASIC 2.0	BASIC 4.0 (lerövidítve)	DOS 5.1.
OPEN – Modus 'A'	APPEND (aP)	
	BACKUP (bA)	
LOAD"$",8&LIST	CATALOG (cA)	@$vagy >$
V(alidate)	COLLECT (coL)	@V vagy >V
	CONCAT (conC)	
C(opy)	COPY (coP)	@C:... vagy >C:...
CLOSE...	DCLOSE(dC)	
LOAD"...",8	DLOAD (dL)	↑file vagy/file
OPEN..,8,....	DOPEN (dO)	
OPEN 1,8,15....	DS$,DS	@ vagy >

BASIC 2.0	BASIC 4.0 (lerövidítve)	DOS 5.1.
SAVE"...",8	DSAVE (dS)	
N(ew)	HEADER (hE)	@N:.. vagy >N:..
I(nitialise)	I(nitialise)	@I vagy >I
P	RECORD (reC)	
R(ename)	RENAME (reN)	@R:.. vagy >R:..
S(cratch)	SCRATCH (sC)	@S:.. vagy>S:..

A fenti táblázat tartalmazza a különböző BASIC-változatokat. A DOS 5.1 rajta van a TEST/DEMO lemezen, leírását pedig a 4.2.1. alfejezetben találja meg az olvasó.
A BASIC 2.0 és a BASIC 4.0 közötti lényeges eltérés az, hogy a BASIC 2.0-nál a lemezműveleteket és a parancsokat a 15. csatornán keresztül kell továbbítani. A BASIC 4.0 lemezműveletei az említett csatornát önállóan kezelik (az INITIALISE kivételével). Ez a BASIC változat pl. a HEADER D0,"DISK1", IHJ utasításból ugyanazt az utasítássorozatot állítja elő, amelyet a BASIC 2.0 változatban a következőképpen kell megadni:

```
OPEN 1,8,15,"N:DISK1,HJ"
CLOSE 1
```

Ismerkedjünk meg részletesebben a BASIC 4.0 utasításokkal!
A magyarázatokban az alábbi jelöléseket használjuk:
- lfn = logikai file-szám
- dn = a meghajtóegység száma – a kettős meghajtóegységeknél DRIVE0 (D0) és DRIVE1 (D1); az egyszeres meghajtóegységek címzése D0-val történik
- ga = egységszám (U4-től U31-ig)

A zárójelek közé írt paraméterek megadása nem kötelező.

APPEND

Az utasítás egyenértékű a BASIC 2.0 "A" módú file-megnyitásával. APPEND-del soros file-t bővíthetünk. Az utasítás formátuma:

```
APPEND# lfn,"file-név" (,Ddn, Uga)
```

A következő példában a 0-s egységben lévő lemezen a SEQU.1 soros file-t bővítjük egy rekorddal

```
100 APPEND#1,"SEQU.1",D0
110 PRINT#1,X$
120 CLOSE 1
```

BACKUP

Ezzel az utasítással teljes lemezt másolhatunk, de csak akkor, ha kettős meghajtóegységgel rendelkezünk. Az utasítás formátuma:

BACKUP Ddn TO Ddn(,Uga)

Fontos, hogy a két egységet – amelyről és amelyre másolunk – mindig megadjuk! Például:

BACKUP D1 TO D0

CATALOG

Az utasítás betölti a lemez tartalomjegyzékét anélkül, hogy a tár aktuális tartalmát felülírná. Formátuma:

CATALOG (Ddn,Uga)

Például a CATALOG D0 utasítás betölti a D0-ban levő lemez tartalomjegyzékét.
Ha nem adunk meg egységszámot, az utasítás hatására kétszeres meghajtónál mindkét lemez tartalomjegyzéke megjelenik a képernyőn.

COLLECT

Azonos hatású a BASIC 2.0 VALIDATE utasításával.
Formátuma:

COLLECT(Ddn)

CONCAT

Soros file-ok összefűzésére szolgál.
Formátuma:

CONCAT(Ddn,)"file1" TO(Ddn)"file2"(ON Uga)

Ha például a D0-ban levő SEQ,2 file-t a D1-ben levő SEQ.1 file mögé akarjuk illeszteni, a következő utasítást kell megadni:

CONCAT D0,"SEQ.2" TO D1, "SEQ.1"

COPY

Ezzel az utasítással file-okat – a relatív file-ok kivételével – másolhatunk egyik meghajtóegységről a másikra. Az utasítás tehát az egyszeres meghajtóegységeknél nem alkalmazható. Formátuma:

COPY (Ddn,)("file1")TO (Ddn,) ("file2")

Ha az összes file-t át akarjuk vinni (például a 0. meghajtóegységről az 1. meghajtóegységre), elegendő az alábbi utasítást végrehajtani:

COPY D0 TO D1

DCLOSE

A DCLOSE utasítás funkciója ugyanaz, mint az egyszerű CLOSE utasításé, az alábbi eltérésekkel:

DCLOSE	lezárja az összes file-t
DCLOSE#1	az 1-es számú file-t zárja le
DCLOSE#1 ON U9	a 9-es egység összes file-ját lezárja
DCLOSE U8	a 8-as egység összes file-ját lezárja

Az utasítás formátuma a következő:

DCLOSE (#lfn) (ON Uga)

DLOAD

A DLOAD utasítás előnye, hogy alapértelmezésben a 8-as egységről végzi a betöltést. Formátuma a következő:

DLOAD "program" (,Ddn) (,Uga)

Ha például a PRG.2 programot akarjuk betölteni a 0 meghajtóegységről vagy egy egyszeres meghajtóegységről, az alábbi utasítást kell végrehajtanunk:

DLOAD "PRG.2"

DOPEN

A BASIC 4.0-nak ez az utasítása rendkívül sokrétű, ami a formátumán is látszik:

DOPEN#lfn,"file" (,Ddn) (,Uga) (,file-paraméter)

Sokoldalúságát elsősorban a választható file-paraméterek adják. Két file-paramétert használhatunk, amelyek jelentését a következő táblázatban foglaltuk össze:

L PARAMÉTER	W PARAMÉTER	FUNKCIÓ
IGEN	NEM	Relatív file megnyitása írásra
NEM	IGEN	Soros file megnyitása írásra
NEM	NEM	Tetszőleges file megnyitása olvasásra (REL, SEQ,PRG,USR)

Az L paraméterhez a rekord hosszúságát is meg kell adni (pl. L80):

DOPEN#1,"FILE.REL",D0,L80

Egy 80 byte-os rekordhosszúságú relatív file-t nyitottunk meg írásra. Ha nem adunk meg file-paramétert, a file megnyitása olvasásra történik.

DS$ & DS

A DS$ és a DS rendszerváltozók, amelyekben a hibaüzenet szövege, illetve száma van elhelyezve.
Lemezhiba után kiírhatjuk a teljes hibaüzenetet PRINT DS$ vagy PRINT DS utasítással.
A hibát programon belül is lekérdezhetjük:

```
100 IF DS=19 THEN GOTO........
```

DSAVE

Programok tárolása lemezen.
Formátuma:

```
DSAVE (Ddn) "programnév" (,Uga)
```

HEADER

A BASIC 4.0 változatban a HEADER utasítással történik a lemezek formálása. Ez a BASIC 2.0 NEW utasításának felel meg.
Az utasítás formátuma a következő:

```
HEADER "lemez neve",D0,lid(,Uga)        vagy
HEADER Ddn, "lemez neve",lid
```

Az egységet kétféleképpen adhatjuk meg. Az lid itt is a lemez azonosítására szolgál. Ha nem adjuk meg, akkor a már megformázott lemezek csak új nevet kapnak és az összes rajtuk levő file törlődik.

RECORD

Ez az utasítás a BASIC 2.0, illetve a DOS 2.6. pozicionálási utasításának felel meg. A RECORD utasítással egy relatív file-ban anélkül is pozicionálhatunk egy rekordra, hogy ezt a pozicionálást továbbítanunk kellene a 15. csatornán keresztül. Formátuma:

```
RECORD#lfn,rn,(,bp)
```

A logikai file-szám a már megnyitott relatív file-ra vonatkozik. Az rn a rekord kulcsát (1-65535), a bp a rekordon belüli pozíciót (1-254) adja meg.
Egy példa: tegyük fel, hogy egy 2-es logikai számú relatív file 128. rekordjának 12. byte-jára akarunk pozicionálni.
A megoldás:

```
RECORD#2,128,12
```

RENAME

Azonos a BASIC 2.0 RENAME utasításával.
Formátuma:

RENAME (Ddn,) "korábbi név" TO "új név" (,Uga)

SCRATCH

Ez a file-törlési eljárás lényegesen kényelmesebb, mint a BASIC 2.0 változat SCRATCH utasítása, mivel egy utasítással elvégezhető. Formátuma a következő:

SCRATCH (Ddn,) "file" (,Uga)

Egy SCRATCH utasítás beírását követő ARE YOU SURE? (biztos vagy benne?) üzenetre adott válasszal érvényteleníthetjük az utasítást. Ha a file-t tényleg törölni akarjuk, az Y-t kell leütni, ha nem, az N-t. A file törlése után a FILES SCRATCHED üzenet jelenik meg a képernyőn.

2. FEJEZET
PROGRAMOZÁS HALADÓK RÉSZÉRE

2.1. A lemez blokkjainak közvetlen elérése

Programozási munkánk során az adatfeldolgozási feladatoknál nem kell foglalkoznunk azzal, hogy fizikailag hogyan történjen az adatállományok elhelyezése a lemezen. Erről a DOS gondoskodik. Igényesebb feladatoknál azonban előfordul, hogy szükségünk van a lemez egy adott blokkjának tartalmára és rendkívül körülményes lenne, ha ezt a feladatot az eddig megismert file-kezelő utasításokkal kellene megoldanunk. A DOS megadja a lehetőséget arra, hogy a lemez elemi adatterületeit, a blokkokat közvetlenül elérjük, azaz tartalmukat beolvassuk, vagy felülírjuk. Ezekhez a műveletekhez egy adatcsatornát és egy puffert kell használnunk. Az adatpuffer feladata az adatok átmeneti tárolása írás vagy beolvasás előtt.
Az OPEN utasításban a # file-név utal arra, hogy közvetlen blokkműveleteket akarunk végezni:

```
OPEN 1,8,2, "#"
```

A fenti utasítással az 1-es logikai file-hoz (a 8-as egységen) hozzárendelünk egy *közvetlen elérésű* (random) file-t. Az adatátvitel a 2-es csatornán keresztül történik. A csatornaszám értékének 2–14 közé kell esnie. A program további részében ezt az adatcsatornát más célra nem használhatjuk, a második alkalmazás pillanatában ugyanis a DOS az elsőt automatikusan lezárja.
Az OPEN végrehajtásakor a DOS keres egy szabad adatpuffert és hozzárendeli az általunk használt adatcsatornát. Az OPEN-t követő GET utasítással lekérdezhetjük ezt a pufferszámot:

```
100 OPEN 1,8,2, "#"
110 GET# 1, A$
120 PRINT ASC (A$+CHR$(Ø))
RUN
3
```

A mi esetünkben tehát a 3. puffer volt szabad. A pufferek számozása 0-tól 4-ig terjed. Ezek egyenként 256 byte-ot foglalnak le (ugyanúgy, mint a lemezen levő blokkok), és a VC 1541-es egység tárában az alábbi tárcímeken helyezkednek el:

PUFFERSZÁM	TÁRTERÜLET
0	$300–$3FF, 768–1023
1	$400–$4FF, 1024–1279
2	$500–$5FF, 1280–1535
3	$600–$6FF, 1536–1791
4	$700–$7FF, 1792–2047

A 4. puffer általában foglalt, mivel a DOS a BAM-ot itt tárolja. Ha file-okkal dolgozunk, a 3. puffer sem használható, mivel ezt a tartalomjegyzék foglalja el. Ha közvetlen lemezműveleteknél egy meghatározott puffertárat akarunk lefoglalni, ezt a következőképpen adhatjuk meg:

```
OPEN 1,8,2, "#3"
```

Az utasítás a 3. puffert rendeli hozzá a 2. csatornához ($600–$6FF), ha az még szabad. Általában jobb, ha a pufferválasztást a DOS-ra bízzuk.
Konkrét pufferválasztásnál le kell kérdeznünk a hibacsatornát:

```
130 OPEN 15,8,15
140 GET#15, A$ : PRINT A$; : IF ST 64 THEN 140
```

Ha a puffer már foglalt, az alábbi hibaüzenetet kapjuk:

```
70, NO CHANNEL, 00, 00
```

Ha nem dolgozunk más file-lal, maximum 4 csatornát nyithatunk meg közvetlen eléréshez.

P. 30

```
10 open 1,8,15,"i0":i=2:rem parancscsatorna
20 open 2,8,2,"#": gosub100
30 open 3,8,3,"#": gosub100
40 open 4,8,4,"#": gosub100
50 open 5,8,5,"#": gosub100
60 open 6,8,6,"#": gosub100
70 end
100 get#i,a$:print asc(a$+chr$(0))
110 i=i+1: rem pufferszam
120 get#1,a$:printa$;:ifst<>64then120
130 return
```

```
 3
00, OK,00,00
 2
00, OK,00,00
 1
00, OK,00,00
 0
00, OK,00,00
 199
70, NO CHANNEL,00
```

A P.30-as program futtatási eredményéből láthatjuk, hogy mi történik az 5. csatorna megnyitásakor.
A közvetlen elérésű utasítások a lemezről beolvasott információt a pufferbe továbbítják, amelyeket innen a szokásos módon INPUT# vagy GET# utasításokkal olvashatunk be a tárba.
Az INPUT# utasítást a pufferben tárolt szövegtípusú változók beolvasására csak akkor használhatjuk, ha hosszuk nem több, mint 88 karakter, CHR$ (13) karakterrel vannak egymástól elválasztva, és ha a szövegben vezérlőkarakterek vagy elválasztókarakterek (vesz-

szó, kettőspont) nem szerepelnek. Ha ezek a feltételek nem teljesülnek, az INPUT# helyett a GET# utasítást alkalmazhatjuk, amely karakterenként végzi a beolvasást.
A karakterenkénti beolvasás során az üres füzért külön kell kezelni:

```
100 GET#2,A$ : IF A$=" " THEN A$=CHR$ (0)
```

A legkényelmesebb megoldás a 4.3.1. alfejezetben ismertetett INPUT* utasítás, amelyben paraméterként megadhatjuk a beolvasandó karakterek számát (ennek értéke maximum 255 byte, tehát majdnem a teljes puffertár), és nem kell külön foglalkoznunk sem az elválasztó karakterekkel, sem a nulla-byte-tal (CHR$ (0)). Az alábbiakban részletesen bemutatjuk az összes közvetlen elérésre alkalmas utasítást és számos példát arra, hogy milyen speciális programozási feladatokban használhatók célszerűen.

2.2. A közvetlen elérésű utasítások

A címben jelzett utasításokkal a lemez tetszőleges adatblokkját közvetlenül kezelhetjük; beolvashatjuk, tartalmát módosíthatjuk. Minden ilyen utasítást a 15-ös parancscsatornán keresztül küldünk az egységhez, tehát a parancscsatornát előzetesen mindig meg kell nyitni.
A közvetlen elérésű utasításokat leggyakrabban a lemezre, ill. a lemezen tárolt file-okra vonatkozó adminisztratív információkkal való manipulálás során alkalmazzuk. Ezeket az információkat a DOS minden lemezen fix helyen, a 18-as sávon tárolja. A 18-as sáv belső szerkezetét a 3.4. fejezetben behatóan ismertetni fogjuk, de mivel a közvetlen elérésű utasítások bemutatására szolgáló mintaprogramok többnyire a 18-as sávon lévő adatblokkokkal dolgoznak, előzetesen közlünk néhány adatot az itt tárolt információkról. Az adminisztrációs bejegyzéseket három alapvető részre oszthatjuk:

1. a lemez adatai
2. a lemez blokkjainak foglaltsági térképe (BAM)
3. a lemezen tárolt file-okra vonatkozó bejegyzések

A lemez adatai és a BAM a 18 sáv 0-s szektorán helyezkednek el, a file bejegyzések pedig a 18-as sáv 1-es szektorán. A lemezen tárolt file-ok adatblokkjait a DOS ún. *láncolás*sal tartja nyilván. Ez azt jelenti, hogy minden adatblokk első két byte-ja az adott file következő adatblokkjának sávját és szektorát tartalmazza. Mivel a 18-as sáv is egy speciális file-ként szerepel, ennek adatblokkjai is a fent leírt módon vannak összeláncolva.

2.2.1. A Block-Read (B–R) utasítás

A Block-Read utasítással beolvashatunk egy blokkot egy előzetesen megnyitott pufferbe. Az utasítás rövidítése: B–R.
A 2.4.4. alfejezetben felsorolt ún. *user utasítás*ok közül az U1 majdnem azonos a B–R utasítással, a különbség mindössze annyi, hogy míg a B–R az olvasást a második byte-tól kezdi, az U1 a teljes adatblokkot beolvassa. Az utasítás formátuma:

U1 Csatornaszám Meghajtóegység Sáv Szektor

A csatornaszám azonos azzal, amit az OPEN-ben megadtunk, a meghajtóegység száma a VC 1541-nél mindig 0. Az alábbi programmal beolvassuk a 18-as sáv 0-s szektorából az első két byte tartalmát:

P. 31

```
10 open 1,8,15
20 open 2,8,2,"#"
30 print#1,"u1 2 0 18 0"
40 get#2, a$,b$
50 printasc(a$+chr$(0)),asc(b$+chr$(0))
60 close2:close1
```

Az eredmény 18 1. Mivel ez a két byte egy adatblokk két első byte-ja, láncolási információt tartalmaz, nevezetesen azt mutatja, hogy a 18-as sávon elhelyezett nyilvántartás következő adatblokkja a 18-as sáv 1-es szektora, ami éppen a tartalomjegyzék első adatblokkja.
A B–R utasítás alkalmazására több mintaprogramot találunk az alábbiakban. A TEST/DEMO lemez DISPLAY T&S programja is ezt az utasítást használja a BAM beolvasására, ill. annak grafikus kijelzésére.
A B–R utasítással beolvasott adatblokk 256 byte-ját GET # utasítással olvashatjuk be a tárba. A P.33. programmal a 18-as sáv 0-s szektorából a lemez nevét és azonosítóját olvassuk be a 144. byte-tól kezdődően.
A fejezet elején leírt láncolási szabályt alkalmazva, B–R utasítással kiírhatjuk képernyőre azokat a sáv- és szektorszámokat, amelyeken egy meghatározott file adatblokkjai találhatóak. Ehhez csak a file első adatblokkjának helyét kell meghatározni (l. 4.1.1. alfejezet), majd az adatblokk első két byte-ját rendre beolvasva a láncoláson végighaladni. A file végét az jelzi, hogy a következő adatblokk sávjainak száma helyett a nulla áll. Ekkor a következő byte megadja, hogy az utolsó sávon hány byte-ot foglal még el a file. Ha az alábbi mintaprogramnak indulásként a 18-as sáv- és a 0-s szektorszámot adjuk, a BAM, ill. a tartalomjegyzék adatblokkjainak sáv- és szektorszámát kapjuk meg.

P.32

```
100 open 1,8,15
110 open 2,8,2,"#"
120 input"Sav es szektor";t,s
130 print#1,"u1 2 0";t;s
140 get#2,t$,s$
150 t=asc(t$+chr$(0)):s=asc(s$+chr$(0))
160 if t=0 then close2:close1:end
170 print t;"sav",s;"szektor"
180 goto 130
```

2.2.2. A Block-Pointer (B–P) utasítás

Ha a lemez nevét és azonosítóját (amely a 18-as sáv 0-s szektorának 144. byte-ján kezdődik) akarjuk beolvasni, az előző módszerrel 143 byte-ot át kell olvasnunk, hogy a névhez hozzájuthassunk. Egy tetszőleges byte elérését megkönnyíti a Buffer–Pointer utasítás. Ezzel az utasítással tetszőlegesen beállíthatjuk azt a *mutatót* (pointer), amely meghatározza az aktuális írás vagy olvasás kezdőpozícióját a pufferben. Az utasítás formátuma a következő:

B–P Csatornaszám Pozíció

Most már közvetlenül olvasható a lemeznév:

P.33

```
100 open 1,8,15
110 open 2,8,2,"#"
120 print#1,"u1 2 18 0"
130 print#1,"b-p 2 144"
140 fori= 1 to 16: rem maximalis hossz
150 get#2,a$:ifa$=chr$(160)then170
160 printa$;:next
170 close2:close1
```

A blokk beolvasása után a puffermutatót 144-re állítottuk, ezután 16 byte olvasása következik, hacsak közben a rendszer nem talált CHR$(160) karaktert (Shift Space), amely a név végét jelenti. A byte-ok a pufferban 0-tól 255-ig vannak számozva, így az első byte a 0. Egy blokknak U1-gyel történő olvasásakor a puffermutató automatikusan a nulladik byte-ra állítódik. A puffermutató mozgatásakor teljesen szabadon járhatunk el. A fenti példánkban például a név olvasása után a 2. byte olvasása egyszerűen megvalósítható, ha a puffermutatót erre a byte-ra állítjuk:

PRINT 1, "B–P 2 2"

2.2.3. A Block–Write (B–W) utasítás

A Block–Write utasítással a puffertár tartalmát beírhatjuk a lemez bármely blokkjába. A Block–Read és Block–Write utasítások együttes használatával bármely blokk tartalmát közvetlenül módosíthatjuk. A B–W utasítással egyenértékű az U2 user utasítás (l. 2.4.4. alfejezet), azzal a különbséggel, hogy a B–W utasítás a kiírási művelet végrehajtása közben a puffermutató pillanatnyi tartalmát beírja a puffer első byte-jába. Ha írás során ezt el akarjuk kerülni, az U2 utasítást kell használnunk. Az utasítás formátuma ugyanolyan, mint a B–R utasításé:

U2 Csatornaszám Meghajtóegység Sáv Szektor

P.34

```
100 open 1,8,15
110 open 2,8,2,"#"
120 print#2,"teszt adat"
130 print#1,"u2 2 0 1 0"
140 close2:close1
```

A TESZT ADAT szöveget először a kettes csatornához tartozó pufferbe, majd a lemez 1. sávjának 0. szektorába írjuk. Az U1 utasítással a puffer tartalma és a puffermutató nem változik.
Nézzük meg, hogyan használhatjuk a Block–Write utasítást a lemez nevének megváltoztatására. Ha az új lemeznév 16 karakternél rövidebb, CHR$ (160) karakterekkel 16 karakterre kell kiegészítenünk, mielőtt felírnánk a lemezre. Ismét a B–P utasítást alkalmazzuk, mellyel a mutatót a pufferen belül a kívánt pozícióra állítjuk:

P.35

```
100 open 1,8,15
110 open 2,8,2,"#"
120 print#1,"u1 2 0 18 0"
130 print#1,"b-p 2 144"
140 a$="lemeznev"
150 if len(a$)<16 then a$=a$+chr$(160):goto 150
160 print#2,a$;
170 print#1,"u2 2 0 18 0"
180 close2
190 print#1,"i0":close1
```

Először beolvassuk a 18. sáv 0. szektorát a pufferba, majd beállítjuk a puffermutatót a lemeznév pozíciójára és beírjuk a 16 karakter hosszú nevet a pufferbe. A 170. sorban a puffer tartalmát visszaírjuk az eredeti blokkba és a 2. csatornát lezárjuk. Végül a lemezt inicializáljuk, hogy a BAM-ot és a nevet a DOS tár átvehesse. Ha most a tartalomjegyzéket betöltjük,

LOAD "$",8
LIST

látható, hogy a lemez új nevet kapott.

2.2.4. A Block–Allocate (B–A) utasítás

A Block–Allocate utasítás feladata, hogy egy blokkot a BAM-ban foglaltnak minősítsen. Erre akkor van szükség, ha közvetlen eléréssel olyan blokkokat írtunk a lemezre, amelyek nem tartoznak egy file-hoz sem, így automatikusan nem is minősülnek foglaltnak. Ha ezt az utasítást nem hajtjuk végre, egy soron következő írás műveletnél a blokkot a DOS lefoglalja és felülírja. A Block–Allocate utasítás formátuma a következő:

B–A Meghajtóegység Sáv Szektor.

Ha a lefoglalandó blokk már foglalt volt, úgy a 65,NO BLOCK hibaüzenetet kapjuk.

P.36

```
100 open 1,8,15
110 input"sav, szektor";t,s
120 print#1,"b-a 0";t;s
130 input#1,a$$,b$,c$$,d$
140 print a$,b$,c$,d$
```

A program bekéri azt a sávot és szektort, amelyeket foglaltnak akarunk minősíteni. Ha a blokk még szabad, a program lefoglalja és a 00, OK,00,00 üzenet jelenik meg. Ha viszont a blokk már foglalt volt, a 65, NO BLOCK,TT,SS üzenetet kapjuk. Az üzenet sáv- és szektorszáma – TT és SS – adja meg az első szabad szektort és sávot. Ha azonban a 65-ös hibaüzenetnél a sáv- vagy a szektorszám egy nullát ad vissza, az azt jelenti, hogy már nincs szabad sáv- és/vagy szektor-számú blokk. Az alábbi program automatikusan mindig a következő szabad szektort foglalja le:

P.37

```
100 open 1,8,15
110 input"sav, szektor";t,s
120 print#1,"b-a 0";t;s
130 input#1,a$,b$,tt,ss
140 if a$="00"then190
150 ifa$<>"65"thenprinta$","b$","tt","ss:end
160 if tt = 0 then print"Nincs tobb szabad blokk":end
170 if tt=18 then tt=19:ss=0
180 t=tt:s=ss:goto120
190 print tt". sav"ss". szektor lefoglalva."
```

A 170. sorban a 18-as sáv lekérdezése megakadályozza, hogy a tartalomjegyzékhez tartozó blokkot foglaljunk le, hiszen ezzel a lemez legfontosabb információterületét rontanánk el. Ha olyan blokkot próbálunk lefoglalni, amely egyáltalán nem létezik, például a 20. sáv 21. szektorát, az alábbi hibaüzenetet kapjuk:

66, ILLEGAL TRACK OR SECTOR,20,21

Ha a BAM-ban egy blokk foglaltnak minősül, akkor a blokkot más file nem tudja felülírni. A blokk egészen addig foglaltnak minősül, amíg a lemez nem kap VALIDATE utasítást (a BASIC 4.0-ban COLLECT). Ez az utasítás egy új BAM-ot állít össze, ami a következőképpen történik. Mivel egy file összes blokkja össze van láncolva, a lemezen a file nyomon követhető. A láncolást a VALIDATE utasítás érvényesíti, azaz minden blokkot foglaltnak jelez, amely egy file-hoz tartozik. A tartalomjegyzékben *-gal jelölt, még le nem zárt file-ok ekkor törlődnek. Így a B–A utasítással lefoglalt, de szabályos file-hoz nem tartozó összes blokkok is újra felszabadulnak. Ha tehát blokkokat foglalunk le, melyek nem tartoznak a tartalomjegyzékben nyilvántartott egyetlen file-hoz sem, a VALIDATE utasítást nem célszerű használni, ilyenkor ugyanis az összes ilyen blokk ismét felszabadul.

2.2.5. A Block–Free (B–F) utasítás

A Block–Free utasítás a Block–Allocate utasítás ellentéte, azaz egy blokkot újra felszabadít a BAM-ban.
Formátuma hasonló a Block–Allocate utasításáéhoz:

B–F Meghajtóegység Sáv Szektor

```
100 OPEN 1,8,15
110 PRINT#1,"B-F 0 20 9"
```

A fenti példában felszabadítjuk a BAM-ban a 20. sáv 9. szektorán levő blokkot. Ha a blokk már szabad volt, nem kapunk hibaüzenetet. A blokkok lefoglalása és felszabadítása nem érinti a Block–Write és a Block–Read utasításokat. A B–W utasítással ugyan felülírhatunk egy blokkot, de az ettől még nem válik foglalttá a BAM-ban. Ha a lemezen csak közvetlen elérésű file-ok vannak, elvileg nem szükséges a leírt blokkokat foglaltnak minősíteni, miután más file-t már nem írunk a lemezre. Ilyen esetben akár a tartalomjegyzék-blokkokat is teleírhatjuk a 18. sávon, s így 672 blokkot használhatunk a VC 1541-es készülék lemezén.

2.2.6. A Block–Execute (B–E) utasítás

A Block–Execute utasítás feladata az, hogy egy blokkot lemezről beolvasson a pufferbe és a puffer tartalmát gépi kódú programként futtassa a DOS-ban. Azokat a rutinokat, amelyeket íly módon akarunk futtatni, a B–W vagy az U2 utasítással felírhatjuk a lemez valamely szektorára, a későbbiek során a Block–Execute utasítás segítségével tölthetjük egy pufferbe, majd ezt gépi programként futtathatjuk. Ehhez természetesen a DOS felépítésének beható ismerete szükséges. A B–E utasítás alkalmazásakor, ha a gépi kódú rutint egy fix területre, azaz egy meghatározott pufferbe akarjuk elhelyezni, a közvetlen elérésű csatorna megnyitásakor meg kell adni a pufferszámot. A Block–Execute utasítás formátuma a következő:

B–E Csatornaszám Meghajtóegység Sáv Szektor

```
100 OPEN 1,8,15
110 OPEN 2,8,2, "#3"
120 PRINT#1, "B–E 2 0 17 12"
```

Itt a 3. puffert ($600–$6FF) a 2. csatornához rendeljük hozzá. Ezután a 17. sáv 12. szektorának tartalmát betöltjük ebbe a pufferbe, s itt programként futtatjuk.
A Block–Execute utasítás a Block–Read és a Memory–Execute utasítással helyettesíthető. A gépi kódú programok DOS-ban történő futtatására vonatkozó példákat a 2.4. fejezetben ismertetjük.

2.3. A közvetlen elérés alkalmazásai

A közvetlen elérésű utasítások alkalmazására számos lehetőség kínálkozik. A BAM-szektoron belüli manipulálásokkal megváltoztathatjuk a lemez nevét vagy az ID-t. A tartalomjegyzék használaton kívüli byte-jait igénybe vehetjük járulékos információk tárolására. A file-oknak adhatunk egy másik nevet, nyomon követhetjük egy file egyes blokkjainak láncolását, s ha szükséges, a saját elképzeléseink szerint eszközölhetünk változtatásokat.
Manipulálhatunk a lemezen levő file-ok típusával. Készíthetünk például soros file-ból program-file-t. A tartalomjegyzéken egy nem lezárt file-t a 7. bit beállításával lezárhatunk (a $02-ből $82 lesz). Az ilyen file-okat a tartalomjegyzékben csillag különbözteti meg a szabályos file-októl. A fenti változtatást követően a csillag eltűnik. Egy file-nak a DOS által figyelembe vett, de utasítással mégsem elérhető tulajdonsága a törléssel szembeni védelem. Ehhez csak a file-típus 6. bitjét kell beállítani (a $82-ből $C2 lesz). A tartalomjegyzékben kiíratáskor a változtatás után egy < jelenik meg. A file most a SCRATCH-csel szemben immunis. Ezzel fontos rendszerprogramokat védhetünk meg a lemezen a spontán törlés ellen. Ezeket az alkalmazási területeket a 4.1. fejezet ismerteti, ahol még egyéb lehetőségekre is kitérünk.
Az említett manipulációk alkalmazása akkor lenne a legkényelmesebb, ha beolvashatnánk egy szektort a lemezről, az megjelenne a képernyőn, majd a szükséges módosítás elvégzése után ismét visszaírhatnánk a lemezre. Ennek a feladatnak a 4.6. fejezetben ismertetett Disk-Monitor tesz eleget. Minden Disk-Monitorral végzett művelet előtt célszerű a lemezünkről másolatot készíteni. Ha ugyanis a tartalomjegyzékben vagy a BAM-ban vétünk hibát, előfordulhat, hogy a lemez egész tartalmát elveszítjük. Ha tévedésből töröltünk egy file-t vagy egy programot a lemezen, bosszantó lenne, ha a teljes programot újra be kellene gépelnünk. Ha a törlés óta még nem írtunk semmit a lemezre, a file-t nagyon egyszerűen megmenthetjük. Egy file törlésekor ugyanis a file adatblokkjai nem törlődnek, csak a file-típus helyére kerül

a tartalomjegyzékben nulla, és a BAM-ban a lefoglalt blokkok felszabadulnak. Ha a file tartalomjegyzék bejegyzését megkeressük és a file-típust ismét beállítjuk (a SEQ-hoz $81-et, a PRG-hez $82-t, az USR-hoz $83-at és a REL-hez $84-et), majd végrehajtunk egy VALIDATE utasítást, hogy a file blokkjai ismét foglaltnak minősüljenek, visszanyerjük az elveszített programot vagy adathalmazt.
A közvetlen elérést felhasználhatjuk olyan saját file-szerkezetek előállítására, amelyeket a DOS nem ismer. Ilyen esetben az új típusú file kezelését saját magunknak kell megszerveznünk, olvasáshoz és íráshoz pedig alkalmazhatjuk a közvetlen elérésű utasításokat. Egy ilyen file-forma például az ISAM-file. Az ISAM-ot az angol Index Sequential Access Method kifejezésből, indexsoros elérésű szervezésnek fordíthatjuk. Az ISAM-file minden rekordja közvetlenül elérhető, ugyanúgy, mint a relatív file esetében. Itt azonban az elérés nem a rekordsorszámon keresztül történik, hanem egy ún. elérési kulcs vagy index alapján. Ez az index nem más, mint a rekord egyik mezője. Ha például egy rekord 5 mezőből áll – családnév, keresztnév, utca, irányítószám és helység – a családnevet elérési kulcsként is definiálhatjuk. Ha ezek után egy olyan rekordot keresünk, amelyben a név „Békés", az ehhez szükséges utasítást a „Békés" rekord olvasásának nevezzük. Nem kell tehát törődni a rekordszámmal, vagy egyéb rendezési kritériummal, s szöveges formában is megadhatjuk, hogy melyik rekord olvasása, megváltoztatása, írása vagy törlése szükséges.
Az ISAM-féle rendszerben az elérési kulcsot többnyire még egyszer különállóan is tárolni kell azokkal az információkkal, amelyek arra utalnak, hogy a rekord a lemezen hol található.
A COMMODORE 64-hez is beszerezhető a MASTER programfejlesztő rendszer, amely az ISAM-hoz hasonló file-szerkezetekkel dolgozik és további lehetőségeket is szolgáltat.

2.4. A DOS közvetlen elérése – a Memory utasítások

A 2.2.6. alfejezetben bemutattuk, hogy miként lehet a programokat a DOS tárba betölteni és ott futtatni. A Memory utasításokkal – melyeket most ismertetünk – a DOS minden byte-ját elérhetjük és a RAM-ban, ill. a ROM-ban tetszőleges programot futtathatunk.
Elérhetjük például a DOS operatív tárát, beolvashatjuk a szabad blokkok számát vagy a lemez nevét a BAM pufferből stb. A DOS RAM-ba történő írással megváltoztathatjuk az állandókat, például az egységszámot, vagy a blokk hibaüzenet előtti olvasási kísérleteinek számát. Rutinokat futtathatunk a DOS táron belül is. Ezek lehetnek a DOS saját rutinjai vagy általunk a puffer tárban elhelyezett programok, amiről már szó esett a Block–Execute utasítás tárgyalásánál.
A Memory utasítások sikeres alkalmazása természetesen feltételezi a 6502-es gépi nyelv és a DOS működési elvének, társzervezésének ismeretét.

2.4.1. A Memory–Read (M–R) utasítás

Ezzel az utasítással a DOS minden byte-ja olvasható. Az utasítást a parancscsatorna közvetíti, s az olvasott byte-ot is a parancscsatornában kapjuk vissza, ahonnan a GET#-tel az információ beolvasható.
Az utasítás formátuma a következő:

```
M–R CHR$(LO) CHR$ (HI)
```

Az LO és a HI a kívánt tárcímek alsó, ill. felső byte-ja. A következő program beolvas egy tárcímet, majd annak tartalmát betölti a DOS-ból:

P.38

```
100 open 1,8,15
110 input"Tarcim        ";a
120 hi=int(a/256)
130 lo=a-256*hi
140 print#1,"m-r";chr$(lo)chr$(hi)
150 get#1,a$
160 print asc(a$+chr$(0))
170 close 1
```

Ha például egy lemezen lévő szabad blokkok számát szeretnénk megtudni, nem kell a teljes tartalomjegyzéket beolvasnunk, csak a megfelelő byte-okat, közvetlenül a DOS tárból.

P.39

```
100 open 1,8,15,"i0"
110 print#1,"m-r";chr$(250)chr$(2)
120 get#1,a$:ifa$="" then a$=chr$(0)
130 print#1,"m-r";chr$(252)chr$(2)
140 get#1,b$:ifb$="" then b$=chr$(0)
150 print asc(a$)+256*asc(b$)" blokk szabad"
160 close1
```

Ezzel a megoldással minden egyes olvasandó byte-hoz M–R utasítást kell végrehajtanunk. Ha egyszerre több byte-ot akarunk beolvasni, az M–R utasításban meg kell adni az olvasandó byte-ok számát (harmadik paraméterként):

M–R CHR$ (LO) CHR$ (HI) CHR$ (SZÁM)

Ezt a lehetőséget felhasználva egy M–R utasítással a lemez nevét is beolvashatjuk a BAM puffer tárából, ha tudjuk, hogy inicializáláskor vagy a file-elérés előtt a BAM a $700-as címtől kezdődően töltődik be a pufferbe.

P.40

```
100 open 1,8,15,"i0"
110 print#1,"m-r"chr$(144)chr$(7)chr$(16)
120 input#1,a$
130 print a$
140 close1
```

A fenti rutinnal a programból ellenőrizhetjük, hogy megfelelő lemezt helyezett-e a felhasználó az egységbe.
Az M–R és az alábbiakban következő M–W utasításokkal megtehetjük, hogy a DOS RAM-területről betöltünk egy DOS rutint a puffer tárba, ott saját elképzelésünk szerint módosítjuk, majd innen futtatjuk.

2.4.2. A Memory–Write (M–W) utasítás

A Memory–Read-del ellentétes utasítás, melynek feladata az, hogy adatokat írjon a DOS tárba. Természetesen csak a DOS-RAM–0-s lap (zero page), a verem (stack), a puffertár, valamint az esetleges input/output területek írhatók felül M–R utasítással. Az utasítás formátuma a következő:

M–W CHR$ (LO) CHR$ (HI) CHR$ (SZÁM) CHR$ (ADAT1) CHR$ (ADAT2)

Itt annyi adatot vihetünk át, amennyit a „szám"-ban meghatároztunk, elméletileg tehát 255-öt. Az input puffer mérete miatt az átvihető byte-ok száma utasításonként 34-re korlátozódik. Az utasítás az egységszám megváltoztatására is alkalmas (l. a DISK ADDR CHANGE programot a 4.2.3. alfejezetben). Az egységszám két tárrekeszt foglal el a 0-s lapon. A $77 címen (119) van az egységszám + $20 (32) LISTEN esetén, vagyis amikor a géptől érkeznek az adatok az egységhez. Az ezt követő címen van az egységszám + $40 TALK esetén, vagyis amikor az adatok érkeznek a géphez. Mivel a címek tárolása külön-külön történik, az adáshoz és a vételhez különböző egységek is használhatók. Az alábbi példában a vevő egység számát 9-re, az adó egység számát 10-re állítottuk be. A géptől érkező információk a 9-es egységhez irányítódnak, a géphez érkező információk a 10-es egységről jönnek.

P.41

```
100 open 1,8,15
110 print#1,"m-w"chr$(119)chr$(0)chr$(2)chr$(9+32)chr$(10+64)
120 close1
140 open1,9,15
150 open2,10,15
160 print#1,"i0"
170 input#2,a$,b$,c$,d$
180 print a$","b$","c$","d$
190 close2:close1
```

Programok ezen a módon nem tölthetők be, ugyanis a DOS arról az egységről kísérli meg a program betöltését, ahonnan a file-név érkezett.
Az egységszámot nyilván akkor kell módosítanunk, ha egy géphez egynél több lemezegységet kötünk, és a másodikat pl. 9-es egységként szeretnénk használni.
A fenti szoftveres változtatás azonban csak addig marad érvényben, amíg nem következik egy RESET (pl. ha kikapcsoljuk a gépet). Ha az egységszámot véglegesen meg akarjuk változtatni, az egységben az elektromos összeköttetést meg kell szakítani (ez egy vékony ónréteg átszakítását jelenti, ami bármikor visszaforrasztható).
Miután a DOS számos paramétere a ROM-ban helyezkedik el, a DOS funkcióit messzemenően változtathatjuk. Például azt a lépéstávolságot, amellyel egy sávon a szektorok foglalttá válnak (a 105-nek megfelelő $69 cím tartalma alapállapotban 100), vagy egy hibaüzenet kialakulása előtt az olvasási kísérletek számát (a $6A cím 106 tartalma általában 5). A paraméterek további címeit a 3.1.2. alfejezet foglalja össze.

2.4.3. A Memory–Execute (M–E) utasítás

Ezzel az utasítással behívhatjuk és futtathatjuk a DOS tárban lévő programot. A programokat RTS-sel (Return from Subroutine, $60) kell lezárni.
Az utasítás formátuma a következő:

M–E CHR$(LO) CHR$(HI)

Itt a LO és HI a gépi rutin kezdőcímének alsó és felső byte-ja. Az M–E utasítással nemcsak a DOS – ROM rutinjait futtathatjuk, hanem az M–W-vel írt, puffertárban elhelyezett saját rutinjainkat is. Példaként nézzük most meg, hogyan hívhatunk le egy hibaüzenetet előállító DOS rutint a ROM-ból. A $EFC9 címen a 72, DISK FULL üzenetet kiíró rutin található. Behívása a következőképpen néz ki:

P.42

```
100 open 1,8,15
110 print#1,"m-e"chr$(201)chr$(239)
120 input#1,a$,b$,c$,d$
130 print a$",",b$",","c$",","d$
140 close1
```

A 110-es sorban a rutin kezdőcímét ($EFC9) felbontjuk alsó és felső byte-ra (239 = $EF, 201 = $C9) és ezt adjuk meg az M–E utasítás paraméterenként. Ezután következik a hibacsatorna lekérdezése és az üzenet kiíratása:

72, DISK FULL, 00, 00

Ha saját programot akarunk futtatni, azt az M–E utasítás előtt be kell írnunk valamelyik puffertárba. Ha a programot gyakrabban használjuk, a puffer tartalmát a lemezen tárolhatjuk, s a továbbiak során egy olyan B–E utasítással futtathatjuk, amely az adott blokk tartalmát beolvassa a pufferbe, majd automatikusan indítja a programot. A DOS rutinjaival próbáljuk meg kiíratni a tartalomjegyzéket a megszokottól eltérő formában (a járulékos paraméterek ugyanolyanok, mint a 4.1.1. alfejezetben leírt programban), majd számoljuk meg, hány file van a lemezen. A rutin elkészítésénél használjuk fel a DOS-listát. Ha megterveztük a tartalomjegyzék új formátumát, a file-bejegyzésekből minden nehézség nélkül meghatározhatjuk a kívánt paramétereket és kialakíthatjuk a kívánt formátumot is.

2.4.4. A User (U) utasítások

A user utasítások lehetőséget adnak a lemeztárban elhelyezett programok futtatására. Az utasítások formátuma a következő:

UX

Itt az X az abc egy tetszőleges betűje A-tól J-ig, vagy egy egész szám 1 és 9 között illetve ":" (a kettőspont a 10-et helyettesíti). Az utasítás behívásakor a rendszer a DOS-ban az alábbi címekre ugrik:

UA	U1	$CD5F	Block-Read helyett
UB	U2	$DC97	Block-Write helyett
UC	U3	$0500	
UD	U4	$0503	
UE	U5	$0506	
UF	U6	$0509	

UG	U7	$050C	
UH	U8	$050F	
UI	U9	$FF01	
UJ	U:	$EAA0	Bekapcsolás-RESET

Az U1 és az U2, valamint az UA és az UB utasítást már ismerjük; ezek a Block-Read és a Block-Write utasításokat helyettesítik. Az U3–U8, illetőleg az UC–UH utasítások az 1280–$500 címtől kezdődően a 2. pufferbe ugranak (l. a 2.1. fejezetet). Amennyiben több utasítást akarunk használni, a 2. pufferben elhelyezhetjük a rutinok címtáblázatát; ha viszont csak egy user utasítás (U3) használatos, a program közvetlenül $500-nál kezdődhet.
Az UJ user utasítás a RESET vektorra ugrik: ezáltal az egység bekapcsolás utáni állapotba (alapállapot) kerül.

P.43

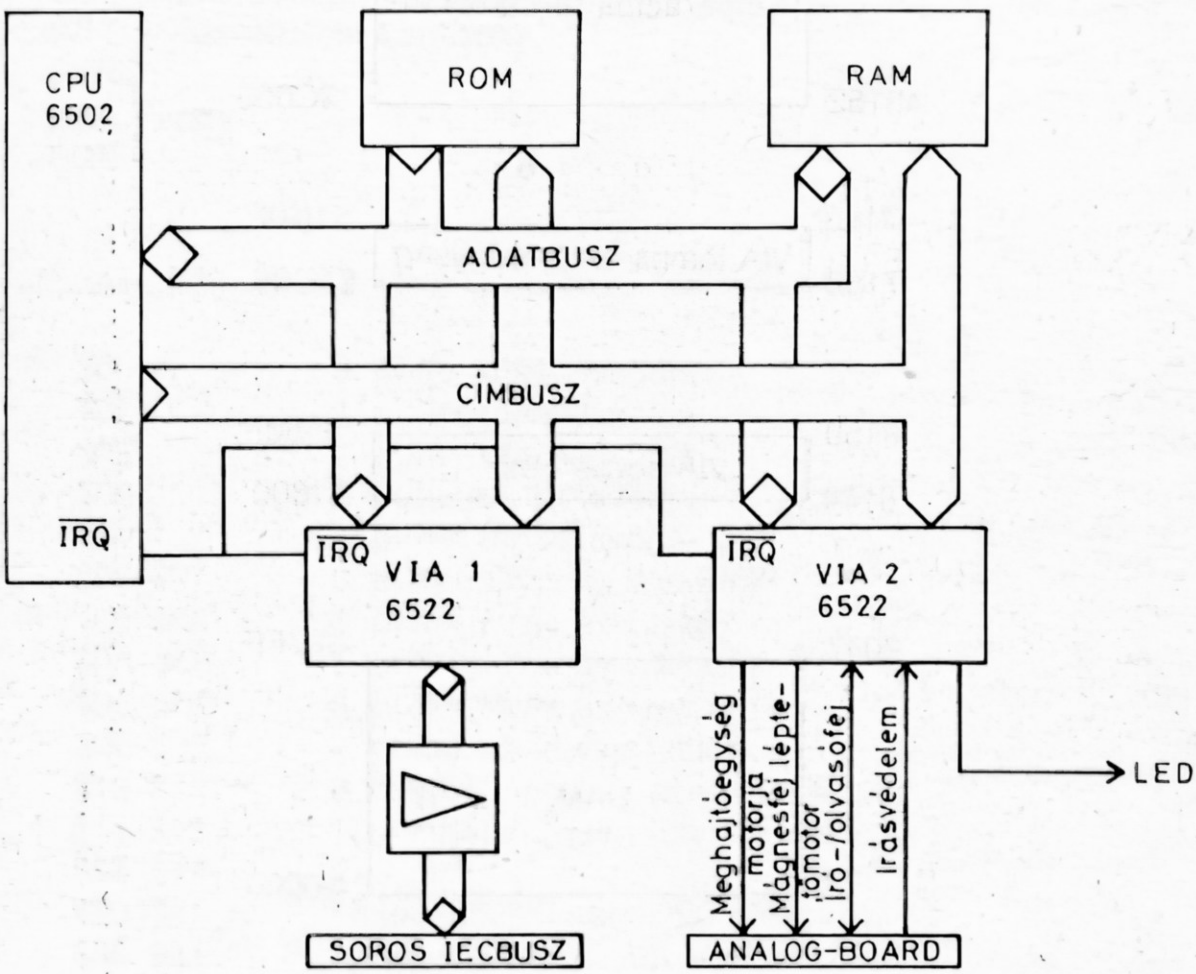
```
100 open 1,8,15
110 print#1,"uj"
120 fori=1 to 1000:next
130 get#1,a$:printa$;:ifst<>64then130
140 close1
```

A 120. sor megvárja az egység RESET-jét. Ezután a 130. sorban beolvassuk az egység bekapcsolási üzenetét.
A user utasítások alkalmazásakor paramétereket is átadhatunk a rutinoknak. Az utasításfüzér az 512-es ($200) címtől kezdődően helyezkedik el az input-pufferben. Itt a paraméterek például a címek, az utasításkódok és a file-nevek lehetnek.
A user utasítások arra is felhasználhatók, hogy a lemezegység utasításkészletét bővítsük vagy egy saját file-szerkezetet valósítsunk meg. Az összes user utasítás helyettesíthető a megfelelő címeket tartalmazó M–E utasításokkal, a user behívás azonban rövidebb és áttekinthetőbb.

3. FEJEZET
A LEMEZEGYSÉG ÉS A LEMEZ MŰSZAKI FELÉPÍTÉSE

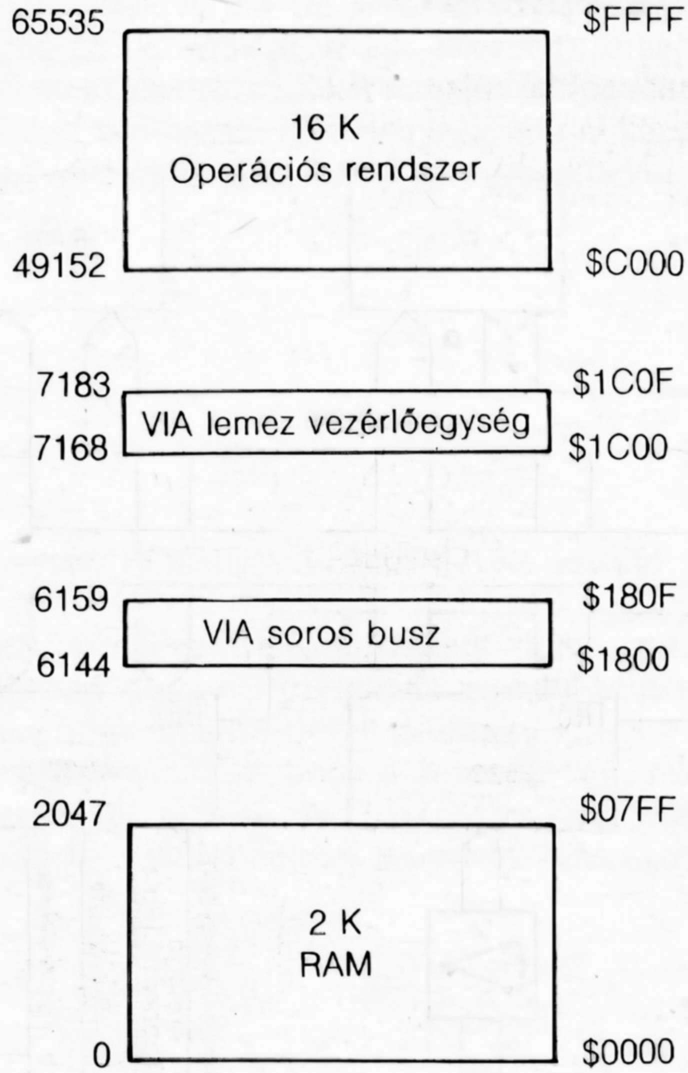
3.1. Az 1541-es felépítése

3.1.1. A lemezegység kapcsolási rajza



3.1.2. A DOS tártérképe (Memory-Map) – ROM, RAM, I/O

A VC 1541-es lemezegység tártérképe:



Az I/O foglaltsága (VIA 6522):

VIA 6522 1, a soros busz kapuja

$1800	B kapu
$1800	A kapu
$1802	B kapu – adatáramlási irány
$1803	A kapu – adatáramlási irány
PB 0:	DATA IN
PB 1:	DATA OUT
PB 2:	CLOCK IN
PB 3:	CLOCK OUT
PB 4:	ATN A
PB 5,6:	egységszám
CB 2:	ATN IN

VIA 6522 2, a motor és az író-/olvasófej vezérlés kapuja

$1C00	B kapu, vezérlőkapu
$1C01	A kapu, adatok az író-/olvasófejtől és az író-/olvasófejhez
$1C02	B kapu – adatáramlási irány
$1C03	A kapu – adatáramlási irány
PB 0:	STP I
PB 1:	STP O – a fejmozgást végző motor
PB 2:	MTR – a meghajtóegység motorja
PB 3:	ACT – LED a meghajtóegységen
PB 4:	WPS – írásvédelem kapcsoló
PB 7:	SYNC
CA 1:	Byte Ready
CA 2:	SOE

A rendszerváltozók tárcímei:

0	$00	0. puffer utasításkódja
1	$01	1. puffer utasításkódja
2	$02	2. puffer utasításkódja
3	$03	3. puffer utasításkódja
4	$04	4. puffer utasításkódja
6	$06–$07	0. puffer sávja és szektora
8	$08–$09	1. puffer sávja és szektora
10	$0A–$0B	2. puffer sávja és szektora
12	$0C–$0D	3. puffer sávja és szektora
14	$0E–$0F	4. puffer sávja és szektora
18	$12–$13	0. meghajtóegység azonosítója (ID)
20	$14–$15	1. meghajtóegység azonosítója (ID)
22	$16–$17	ID
32	$20–$21	Fejtovábbítási kapcsoló (flag)
48	$30–$31	A vezérlés puffermutatója
57	$39	8 állandó fejléc (header) kezdete
58	$3A	Az adatpuffer paritása
61	$3D	Egységszám a lemezvezérlő (Disk-controller) számára
63	$3F	A pufferszám a lemezvezérlő számára
67	$43	Sávonkénti szektorok száma formátumszervezéskor
71	$47	A 7 állandó adatblokk kezdetének jelölése
73	$49	Stackmutató (veremmutató)
74	$4A	Fejtovábbítás lépés-számlálója
81	$51	Aktuális sávszám formátumszervezéskor
105	$69	Olvasási kísérletek száma (5)
106	$6A	Lépéstávolság szektorhozzárendeléskor (10)
111	$6F–$70	Címmutató az M- és a B-utasításnál
119	$77	Egységszám + $20 LISTEN-nél
120	$78	Egységszám + $40 TALK esetén
121	$79	LISTEN-kapcsoló (1/0)
122	$7A	TALK-kapcsoló (1/0)

124	$7C	ATN-kapcsoló vétele a soros buszról
125	$7D	EOI-kapcsoló soros busza
127	$7F	Meghajtóegység-szám
128	$80	Sávszám
129	$81	Szektorszám
130	$82	Csatornaszám
131	$83	Másodlagos cím
132	$84	Másodlagos cím
133	$85	Adatbyte
139	$8B–$8D	Operatív tár osztáshoz
148	$94–$95	Aktuális puffermutató
153	$99–$9A	0. puffer címe – $300
155	$9B–$9C	1. puffer címe – $400
157	$9D–$9E	2. puffer címe – $500
159	$9F–$A0	3. puffer címe – $600
161	$A1–$A2	4. puffer címe – $700
163	$A3–$A4	Az input-puffer mutatója ($200)
165	$A5–$A6	A hibaüzenet-puffer mutatója ($2D5)
181	$B5–$BA	# lo rekord, alsó (LO) blokkok száma
187	$BB–$C0	# hi rekord, felső (HI) blokkok száma
193	$C1–$C6	Relatív file írásmutatója
199	$C7–$CC	Rekordhosszúság relatív file-oknál
212	$D4	Rekordon belüli mutató relatív file-nál
213	$D5	Oldalszektor-szám
214	$D6	Az oldalszektorban lévő adatblokkon belüli mutató
215	$D7	A relatív file adatblokkjának mutatója
231	$E7	File-típus
249	$F9	Pufferszám
$100–$145	256–325	Sáv
$200–$228	512–552	Utasításfüzér-puffer
$24A	586	File-típus
$258	600	Rekordhosszúság
$25A	601	Oldalszektor sávja
$274	602	Oldalszektor szektora
$278	628	Input sor hosszúsága
$278	632	File-nevek száma
$297	663	File üzemmód
$280–$284	640–644	Egy file sávja
$285–$289	645–649	Egy file szektora
$2D5–$2F9	725–761	Hibaüzenet-puffer .
$2FA/$2FC	762/764	Szabad blokkok száma
$300–$3FF	768–1023	0. puffer
$400–$4FF	1024–1279	1. puffer
$500–$5FF	1280–1535	2. puffer
$600–$6FF	1536–1791	3. puffer
$700–$6FF	1792–2047	4. puffer

3.2. A DOS működési elve

A VC 1541-es készülék intelligens lemezmeghajtó egység, amely saját mikroprocesszorral és operációs rendszerrel (DOS, *D*isk *O*perating *S*ystem) rendelkezik. A DOS-nak nincs szüksége az alapgép tárkapacitására és nem veszi igénybe az alapgép műveleti idejét sem. A gép feladata csak annyi, hogy közvetíti az egységhez a felhasználó utasításait, amelyeket aztán az utóbbi maga hajt végre.

A fentiek miatt a DOS-ra egyidejűleg három feladat hárul:

1. Biztosítania kell az adatforgalmat a géphez és megfordítva.
2. Az utasításokat értelmeznie kell, irányítania kell a file-ok, az ezekhez rendelt átviteli csatornák és a blokkpuffer kezelését.
3. Le kell bonyolítania az egység hardver-műveleteit. Idetartozik az egyes blokkok felírása a lemezre, visszaolvasásuk, valamint a lemezek formálása.

A VC 1541-es készüléknél a fenti feladatokat a 6502-es mikroprocesszor végzi el. A feladatok látszólagos egyidejű elvégzése azonban csak a megszakítás-technika (interrupt) segítségével lehetséges.

A DOS főprogramja gondoskodik a közölt utasítások értelmezéséről és futtatásáról. Az adatok és az utasítások átvitele megszakítások formájában zajlik. Ha a gép egy perifériát akar igénybevenni, úgy az ATN vezetéken keresztül egy impulzust küld (ATN = attention, figyelem – l. az 5.1. fejezetet). Ez a külső egységen megszakítást idéz elő. Az egység megszakítja a futó programját és megjegyzi, hogy a gép adatokat akar továbbítani. Ezután dolgozza fel az eredeti utasítást. Ezt követően az egység további adatokat és utasításokat fogadhat el a géptől, s ezeket feldolgozhatja. A kapott utasítások feldolgozása után az egység várakozó ciklusban van, amíg nem érkezik új utasítás a géptől.

Az utasításfeldolgozás ezen a szinten azonban csak az utasítás logikai feldolgozására, a géptől érkező és a gép felé irányuló átviteli csatornák kezelésére és a feladathoz rendelt puffertárba írandó vagy olvasandó adatok behozására korlátozódik. A lemezvezérlő (Disk Controller) feladatait – mint pl. lemezek formálása, valamint az egyes blokkok tényleges írása és olvasása – szintén a processzor látja el.

A lemezműveletek elvégzése is megszakításvezérléssel történik. Egy beépített óra (Timer) kb. 14 milliszekundumonként megszakítja a lemezegység szabályos programját és az egységet egy olyan programhoz ágaztatja el, amely a lemezvezérlő feladatait végrehajtja. A két önálló program közötti kommunikáció azokon a közösen használt tárrekeszeken keresztül valósul meg, ahová a főprogram a lemezvezérlő program utasításkódjait elhelyezi. Amikor a megszakítás program aktív, ezekben a tárrekeszekben megvizsgálja, hogy szükséges-e valamilyen műveletvégzés, például lemezformálás. Ha igen, működésbe lépnek a meghajtóegység- és mágnesfejmotorok. Miután a megszakítás rutin véget ért, a főprogram ismét ellenőrzi a közösen használt tárterületen lévő információk alapján, hogy a lemezvezérlő végrehajtotta-e a feladatot, vagy további várakozás szükséges. Hasonló módon a főprogram arról is értesül, hogy valamilyen hiba lép fel, például a Read Error, vagy írás közben a DOS írásvédelmi tapaszt érzékelt. Így a főprogram megfelelően reagálhat, például előkészíthet egy hibaüzenetet. A nagy CBM lemezegységnél a lemezvezérlő feladatát egy saját, második 6504 típusú mikroprocesszor látja el. A kommunikáció ebben az esetben is a közös tárrekeszeken belül történik. A DOS tártérképét, valamint a lemez és a soros busz kezeléséhez az I/O elemek áttekintését az előző fejezet tartalmazza. A DOS működésének ez a rövid áttekintése természetesen csak vázlatosan ismerteti a funkciókat. Amennyiben az Olvasó behatóbban kíván tájékozódni, elemezze a VC 1541-es egységnek a 3.5. fejezetben ismertetett DOS listáját, amely részletesen dokumentálja a teljes 16 K-s operációs rendszert.

3.3. A VC–1541-es lemezformátuma

Az egységen használt lemez 35 sávra, a sávok pedig 17–21 szektorra vannak felosztva. A szektorok össz-száma 683.
Miután a tartalomjegyzék és a BAM az egész 18-as sávot lefoglalja, 664, egyenként 256 byte-os blokk áll rendelkezésre az adatok és programok tárolására.
A sávok felosztása a következő:

SÁV	SZEKTOROK SZÁMA
1-től 17-ig	21
18-tól 24-ig	19
25-től 30-ig	18
31-től 35-ig	17

A szektorok sávonkénti eltérő számának magyarázata, hogy a középpont felé a sávok rövidülnek.

3.3.1. A VC–1541-es BAM-ja

A BAM (*B*lock *A*vailability *M*ap) feladata, hogy a blokkok foglaltságát nyilvántartsa. Minden blokkművelet után (tárolás, törlés stb.) a BAM aktualizálódik. Ha a BAM alapján megállapítható, hogy egy tárolandó file több blokkot igényel, mint amennyi rendelkezésre áll, akkor egy hibaüzenet kiírására kerül sor. Egy file megnyitásakor az átviteli utasítások aktualizálásával párhuzamosan a BAM-ot átveszi a DOS tár, majd a file lezárásakor visszaírja a lemezre. Az írás vagy törlés funkciójú utasítások a BAM-ot olvassák, aktualizálják és visszaírják. A 18-as sáv 0. szektorának a felépítése a következő:

18-as sáv 0. szektora

BYTE		TARTALOM	JELENTÉS
0,1	($00–$01)	$12,$01	a tartalomjegyzék első blokkjának sávja és szektora
2	($02)	$41	"A" ASCII-karakter; az 1541-es formátumára utal
3	($03)	$00	a jövőbeni használathoz szükséges 0-kapcsoló
4–143	($04–$8F)		foglalt és szabad blokkok térképe (a 35 sáv ábrázolásához éppen 35* 4 = 140 byte-ra van szükség)

A blokkok térképe úgy van kialakítva, hogy minden 4 byte jelöl egy sávot. Mint az alábbi táblázatból is látható, az első byte a sáv szabad blokkjainak számát tartalmazza. A fennmaradó 3 byte 24 bitjén helyezkednek el a sáv blokkjainak foglaltsági mutatói (minden blokknak megfelel egy bit).
Egy sáv BAM bejegyzésének szerkezete a következő:

BYTE	TARTALOM
0	A sáv szabad blokkjainak száma
1	0–7. szektorok bittérképe
2	8–15. szektorok bittérképe
3	16–23. szektorok bittérképe

A következő táblázat a lemez 1. sávjához rendelt 4 byte tartalmát mutatja.

18-as sáv 0. szektora, 4–7. byte (1. sáv)			
00001010 ($0A)	00000000 ($00)	00000011 ($03)	11111111 ($FF)
10 szabad blokk összesen	1 = szabad a blokk 0 = foglalt a blokk		

Egy olyan programciklussal, amely minden 4 byte-os sávbejegyzésből beolvassa az első byte tartalmát és ezeket összegzi, meghatározhatjuk a lemezen levő szabad blokkok számát.

3.3.2. A tartalomjegyzék fejléce

A tartalomjegyzék fejléce az alábbi információkat tartalmazza:
- a lemez neve,
- a lemez azonosítója (ID)
- a DOS-változat száma,
- a file neve
- a file típusa
- a file-onkénti blokkok száma
- szabad blokkok.

A fenti adatokat LOAD "$",8 paranccsal betölthetjük a tárba, majd LIST paranccsal kiírathatjuk a képernyőre.
A BAM és a tartalomjegyzék a lemez 18-as sávját teljesen lefoglalja. Az első adatblokkon (Ø-s szektor) található a BAM és a tartalomjegyzék fejléce, azután következnek a file-bejegyzések. Minden további blokk legfeljebb 8 file-bejegyzést tartalmazhat. Miután a BAM és a tartalomjegyzék fejléce 1 blokkot foglal le, a file-bejegyzésekhez még 18 blokk áll rendelkezésre. Így egy lemezen maximálisan 144 file kezelhető (18 blokk, blokkonként 8 bejegyzés). A 18-as sáv 0-s szektor első 143 byte-jának tartalmát már bemutattuk (ez a BAM).
A tartalomjegyzék fejlécének formátuma a következő:

18. sáv 0. szektora

BYTE		TARTALOM	JELENTÉS
144–161	$90–$A1		a lemez neve SHIFT SPACE-szel kiegészítve
162, 163	$A2,$A3		a lemez azonosítója (ID)
164	$A4	$A0	SHIFT SPACE
165, 166	$A5,$A6	$32, $41	"2A" ASCII karakter (formátum)
167–170	$A7–$AA	$A0	SHIFT SPACE
171–255	$AB–$FF	$00	használaton kívüli nullákkal kitöltve*

A lemez neve

A lemez nevét (amely maximum 16 karakterből állhat) formáláskor kell megadni. 16-nál kevesebb karakter esetén a fennmaradó részt a DOS SHIFT SPACE-szel ($A0) tölti ki.
Az alábbi BASIC rutin beolvassa a lemez nevét:

P.44

```
100 open 15,8,15,"i0":              rem utasitascsatorna nyitas es inicializalas
110 open 2,8,2,"#":                 rem adatcsatorna
120 print#15,"b-r";2;0;18;0:        rem 18 sav 0 szektora olvasva
130 print#15,"b-p";2;144:           rem buffer-pointer a 144. byte-ra allitva
140 dn$="" :                        rem fuzer torlese
150 :                               rem 16 byte beolvasas (lemeznev)
160 :     :fori=1to16
170 :     :get#2,x$  :              rem egy byte olvasasa
180 :     :if asc(x$)=160 then 200: rem shift+space utan nem kell olvasni
190 :     :dn$=dn$+x$:              rem byte a dn$-be
200 nexti
210 close2:close15:                 rem csatornak zarasa
```

A fenti rutin lefutása után a lemez neve a DN$ füzérben lesz.

A lemez azonosítója (ID)

A lemez azonosítóját, mely két karakterből áll, szintén formáláskor kell megadni. A DOS ennek alapján veszi észre a lemezcserét és tölti be az új lemez BAM-ját; ez tulajdonképpen az inicializálás. Miután a DOS az aktuális BAM-ot mindig a tárban tartja, ha véletlenül két lemez ID-je azonos, nem veszi észre a lemezcserét, nem inicializál automatikusan és ez súlyos hibához vezethet (az egyik lemez blokkfoglaltsági térképét ráírja a másikra!)

3.3.3. A tartalomjegyzék formátuma

A 18-as sáv 1-től 19-ig terjedő blokkjai tartalmazzák a file-bejegyzéseket. Ezen belül egy blokk első két byte-ja a következő file-bejegyzéseket tartalmazó blokkra mutat. Ha a file-bejegyzéseknek vége, akkor ez a két byte $00-t és $FF-et tartalmaz.

* A 180-tól 191-ig terjedő byte-ok egyes lemezeken a BLOCKS FREE-t tartalmazzák.

18-as sáv 1. szektora

BYTE		TARTALOM
0,1	($00,$01)	A tartalomjegyzék következő blokkjának sávja és szektora
2–31	($02–$1F)	1. file bejegyzése
34–63	($22–$3F)	2. file bejegyzése
66–95	($42–$5F)	3. file bejegyzése
98–127	($62–$7F)	4. file bejegyzése
130–159	($82–$9F)	5. file bejegyzése
162–191	($A2–$BF)	6. file bejegyzése
194–223	($C2–$DF)	7. file bejegyzése
226–255	($E2–$FF)	8. file bejegyzése

A file-bejegyzés formátuma

Minden egyes file-bejegyzés 30 byte-ból áll, amelyek funkcióit a következő táblázat foglalja össze:

BYTE		TARTALOM
0	($ØØ)	File-típus
1,2	($Ø1,$Ø2)	Első adatblokk sáv- és szektorcíme
3–18	($Ø3–$12)	File-név (SHIFT SPACE-szel kiegészítve)
19,20	($13,$14)	Csak relatív file-oknál használatos az első oldalszektorblokk sáv- és szektorcíme
21	($15)	Csak relatív file-oknál használatos (rekordhosszúság)
22–25	($16–$19)	Használaton kívül
26,27	($1A–$1B)	Az új file sáv- és szektorcíme @-val történő felülírásnál
28,29	($1C–$1D)	Blokkok száma a file-ban (Low-byte, High-byte)

A file-típus jelölése

A file-bejegyzés 0. byte-ja adja meg a file-típust. A DOS file-típussal dolgozik, ezek megkülönböztetésére szolgál a 0. byte utolsó három bitje. A 7. bit arra utal, hogy szabályosan történt-e a file lezárása. Egy file megnyitásakor a típusnak megfelelően a DOS beállítja a 0. byte tartalmát, majd a file lezárásakor a 7. bitet. A nem lezárt file-t a kilistázott tartalomjegyzékben a file-típus előtt egy csillag jelöli. Ha megnyitunk egy TESZT nevű soros file-t, majd betöltjük és kiíratjuk a tartalomjegyzéket, a file a következőképpen jelenik meg:

```
12    "TESZT"    *SEQ
```

Ha a file-t lezárjuk, a tartalomjegyzék következő kiírásánál a csillag már nem jelenik meg. Ha a file-t elfelejtjük lezárni és a későbbiekben meg akarjuk nyitni, a WRITE FILE OPEN hibaüzenetet kapjuk.

A file-típus

Az alábbi táblázat összefoglalja az összes file-típust, a 0. byte megfelelő jelentését:

FILE-TÍPUS	BITMASZK (nyitva)			BITMASZK (zárva)		
	7654	3210	HEX	7654	3210	HEX
DELeted (törölt)	0000	0000	$00	1000	0000	$80
SEQuential	0000	0001	$01	1000	0001	$81
PRoGram	0000	0010	$02	1000	0010	$82
USeR	0000	0011	$03	1000	0011	$83
RELative	0000	0100	$04	1000	0100	$84

A táblázat alapján úgy tűnik, hogy a 0. byte 3-tól 6-ig terjedő bitjeinek nincs funkciója. A DOS-listából azonban kiderül, hogy ez nem így van.
A file-típus 6-os bitje védett file-ra utal.!
Ha ezt a bitet 1-re állítjuk, a file a szokásos módon nem törölhető. Arra, hogy a file védett, a tartalomjegyzékben a file-típus után egy < karakter utal.
A könyv 4. fejezete tartalmaz egy programot, amely segítségével lehetővé válik a file-ok védelme, felszabadítása és törlése.

Az első adatblokk sávja és szektora

A file-bejegyzés 1. és 2. byte-ja a file első adatblokkjának sáv- és szektorszámát adja meg. Az 1. byte a sáv, a 2. pedig a szektor száma. A blokkláncolás úgy történik, hogy az adatblokk első két byte-ja mutat a második adatblokkra és így tovább. Az utolsó adatblokkot arról lehet felismerni, hogy első byte-jának tartalma $00. A második byte az ezen az utolsó blokkon levő, de még a file-hoz tartozó byte-ok számát tartalmazza (alsó byte, felső byte).

A láncolási folyamatot jól szemlélteti a könyvben közölt Disk-Monitor program:

```
>:B0  A0 A0 A0 A0 A0 00 00 00   ......
>:B8  00 00 00 00 00 00 0B 00   ......
>:C0  00 00 81 13 09 54 31 32   ...T12
>:C8  2F 53 30 6  A0 A0 A0 A0   /S01
>:D0  A0 A0 A0 A0 A0 00 00 00   ...
>:D8  00 00 00 00 00 00 06 00   ......     file-típus
>:E0  00 00 [82] [10] [00] 44 49 53   ...DIS     az I. adatblokk sávja
>:E8  4B 20 41 44 44 52 20 43   K ADDR C   az I. adatblokk szektora
>:F0  48 41 4E 47 45 00 00 00   HANGE...
>:F8  00 00 00 00 00 00 [04] [00]   ........   blokkok száma
```

Ez egy kivonat a TEST/DEMO lemez tartalomjegyzékéből (18-as sáv 1. szektor). Kövessük most nyomon a DISK ADDR CHANGE file szervezését. Az említett file bejegyzése $E2 byte-tal kezdődik és $FF byte-tal végződik. Ez egy PRG-file, ami a $E2 byte-ban a $82 file-típusról ismerhető fel. Ez a file 4 blokkot foglal el a lemezen, amit a $FE és a $FF byte-ok mutatnak. A bejegyzés $E3 és $E4 byte-ja a file első adatblokkjának címét adja (a $10, $00 a 16-os sáv 0. szektorának felel meg).

A file tehát a 16. sáv 0. szektorán kezdődik:

```
>:00  [10 0A] 01 04 0F 04 64 00  .......$.
>:08  97 35 39 34 36 38 2C 31    .59468,1      a 2. adatblokk címe
>:10  32 00 39 04 6E 00 99 22    2.9.....”
>:18  93 13 11 11 11 11 44 52    .......DR
>:20  49 56 45 20 41 44 44 52    IVE ADDR
>:28  45 53 53 20 43 48 41 4E    ESS CHAN
>:30  47 45 20 50 52 4F 47 52    GE PROGR
>:38  41 4D 22 00 59 04 6F 00    AM”.Y./.
>:40  99 22 11 54 55 52 4E 20    .”.TURN
>:48  4F 46 46 20 41 4C 4C 20    OFF ALL
```

Ez a blokk tartalmazza a program első részét, amely ebben a formában nehezen olvasható. A BASIC programot a DOS a lemezen ugyanolyan formában tárolja, mint ahogyan a gép tárában. A BASIC utasításokat 1 byte-os kódokkal ún. *token*ekkel helyettesíti. Ebben az ábrázolásban csak a szövegek ismerhetők fel. Az első adatblokk első két byte-ja a második adatblokkra mutat ($10 és $0A, vagyis a 16-os sáv 10. szektora), melynek kivonata a következő:

```
>:00  [10 14] 34 30 00 1D 05 A0  ..40...
>:08  00 8D 20 33 30 30 3A 20    ..300:        a 3. adatblokk címe
>:10  8F 20 46 49 4E 44 20 44    . FIND D
>:18  52 49 56 45 20 54 59 50    RIVE TYP
>:20  45 00 39 05 AA 00 8D 20    E.9. ..
>:28  36 30 30 3A 20 8F 20 43    600: . C
>:30  48 41 4E 47 45 20 41 44    HANGE AD
>:38  44 52 45 53 53 00 68 05    DRESS.<.
>:40  B4 00 99 22 11 54 48 45    ..”.THE
>:48  20 53 45 4C 45 43 54 45     SELECTE
```

A program ebben a blokkban folytatódik. A $00 és a $01 byte most a file 3. adatblokkjára mutat ($10, $14, a 16-os sáv 20-as szektora):

```
>:00  [10 08] 31 30 30 30 00 23  ..1000.#
>:08  06 54 01 8B 20 43 B2 32    .T.. C 2      a 4. adatblokk címe
>:10  35 34 20 A7 20 4D 54 B2    54 MT
>:18  31 31 39 3A 20 8F 3A 20    119: .:
>:20  32 30 33 31 20 56 32 2E    2031 V2.
>:28  36 00 45 06 5E 01 8B 20    6. E. ..
>:30  43 B2 32 32 36 20 A7 20    C 226
>:38  4D 54 B2 35 30 3A 20 8F    MT 50:
>:40  3A 20 32 30 34 30 20 56    : 2040 V
>:48  31 2E 32 00 67 06 68 01    1.2. .(.
```

Ez a program utolsó előtti blokkja. Az Olvasó már biztosan észrevette, hogy bár az adatblokkok ugyanazon a sávon helyezkednek el, mégsem egymás után következnek. Ez azonban nem jelenti a blokkok minden rendszer nélküli lefoglalását. Az első adatblokk a 0. blokk, a következő a 10. blokk, tehát a számolás 10 blokkonként folytatódik, mindig 9 blokk marad

ki. A későbbiekben látni fogjuk ennek a látszólag nehézkes szervezésnek az értelmét. A 3. adatblokk a 20-as blokk. A láncolás ismét az első blokknál folytatódik, ha a 10-zel növelt blokkszám túllép az adott sávon! Miután a 16-os sávon 21 blokk van, az utolsó adatblokk a sáv 8-as blokkja, amelyet a 3. blokk első két byte-ja mutat.

```
>:00  [00] [F8] 5A  42  B2  31  20  A7  . ZB 1
>:08  20   34   34  30  00  14  07  AE  440...
>:10  01   8B   20  53  54  20  A7  20  .. ST
>:18  31   30   30  30  00  45  07  B8  1000.E.
>:20  01   98   31  35  2C  22  4D  2D  ..15,"M-
>:28  52   22   C7  28  31  37  32  29  R"(172)
>:30  C7   28   31  36  29  3A  A1  23  (16):#
>:38  31   35   2C  5A  43  24  3A  5A  15,ZC$:Z
>:40  C7   28   30  29  29  00  66  07  C f( ZC$
>:48  C7   28   30  29  29  00  66  07  g (00).&.
```

az utolsó blokk jelölése

az utolsó blokk file-hoz tartozó byte-jainak száma

Itt a program végét a $00 byte $00 értéke jelzi. A $01 byte adja meg azoknak a byte-oknak a számát, amelyeket a program ebben az utolsó blokkban lefoglalt (a $F8 248 byte-nak felel meg). Most már könnyen meghatározható a program nagysága:

3, egyenként 254 byte-ot tartalmazó blokk	=	762 byte
utolsó blokk	=	248 byte
a program nagysága:		1010 byte

A file-név

A file nevét a file-bejegyzés 3–18. byte-jai tartalmazzák. Ha a file neve kevesebb 16 karakternél, az üres helyeket $A0 (SHIFT SPACE) karakter tölti ki.

Az új file sávja és szektora "átírásnál"

Ha tárolásnál a file-név elé @ jelet teszünk, a DOS a file-t felülírja. Valójában ez nem felülírás, mert először elhelyezi az új file-t a lemez szabad blokkjain, és mivel a file ilyen néven már létezik, nem készít új tartalomjegyzék bejegyzést. Az új file első blokkjának címét a bejegyzés 26. és 27. byte-ja tárolja. Az új program tárolása után a korábbi program törlődik és az eddig lefoglalt blokkok felszabadulnak a BAM-ban. Az új file első adatblokkjának címe betöltődik a bejegyzés 1. és 2. byte-jába, vagyis a file első adatblokkjának címébe, s ezzel megtörtént a file "felülírása".

A file-ban levő blokkok száma

A file-bejegyzés 28. és 29. byte-jában van megadva a file hosszúsága blokkokban. Egy file legalább 1 és legfeljebb 644 blokkot foglalhat le. Az első byte az alsó byte, vagyis a 2 byte-os szám jobboldali része. A másik byte a felső byte.

3.4. A relatív file-ok szervezése

A relatív file-ok szervezése során a rekordok közvetlen eltérését a DOS egy segédfile létrehozásával oldja meg, amely az egyes adatblokkok lemezcímét tartalmazza. Ezt a segéd-file-t a továbbiakban *oldalszektor*nak nevezzük.
Az oldalszektorokat a rendszer minden relatív file létrehozásakor automatikusan kialakítja, ezzel a programozónak nem kell foglalkoznia.

A következő feladaton végigkövethetjük a relatív file szerkezeti felépítését. Nyissunk meg egy 100 karakter hosszú rekordokból álló relatív file-t.

```
OPEN 2,8,2, "RELATIV FILE,L," + CHR$(100),
```

majd írjuk bele a 70. rekordot:

```
OPEN 1,8,15
PRINT# 1, "P" + CHR$(2) + CHR$(70) + CHR$(0) + CHR$(1)
PRINT# 2, "70.REKORD"
CLOSE 2 : CLOSE 1
```

A file-bejegyzés tárkivonata a program lefutása után:

```
>:00   ..   ..   84   11   00   52   45   4C   ...RELATIV
>:08   2D   44   41   54   45   49   A0   A0   -FILE
>:10   A0   A0   A0   A0   A0   11   0A   64        ..$
>:18   00   00   00   00   00   00   1D   00   ........
```

Az első byte a file-típus ($84), a következő két byte ($11, $00; 17-es sáv, 0. szektor) pedig a tényleges adatterület első sávjára és szektorára utal, pontosan, mint egy soros file esetén. Ezt követi a file neve (16 karakter, SHIFT SPACE-szel ($A0) kitöltve. A következő három byte csak a relatív file-nál használatos. Az első két byte az első oldalszektor-blokk helyét mutatja (sáv, szektor).

Az oldalszektor, mint már említettük, a tényleges adatblokkok címét tartalmazza, ennek belső felépítését részletesen megvizsgáljuk ($11, $0A; 17-es sáv 11-es szektor). A következő byte tartalmazza a rekordok hosszúságát, nevezetesen egy 1 és 254 közötti értéket, amely a mi esetünkben $64 = 100. A fix hosszúságra feltétlenül szükség van, mert a DOS ebből számítja ki, hogy a rekordok helyét melyik oldalszektorban kell keresni. A file-bejegyzés többi byte-jának jelentése ismét a szokásos; a két utolsó byte a file által lefoglalt blokkok számát adja meg (alsó és felső byte, $1D és $00 vagyis 29).

Az oldalszektor felépítése és feladata

A közvetlen elérés a következőképpen valósul meg. Ha például a 70. rekordot akarjuk beolvasni a relatív file-ból, a DOS megkeresi az oldalszektorban, hogy a rekord melyik sávon és szektorban van, majd a megtalált címről beolvassa az adatblokkot (soros tárolásnál a láncolás miatt ehhez végig kellene olvasni az első 69 rekordot). Most egy rekord eléréséhez csak néhány blokk olvasására van szükség. A fentiekben vázolt, de egy kissé leegyszerűsített ábrázolás után nézzük meg most közelebbről egy oldalszektor pontos felépítését.

```
>:00   00   47   00   64   11   0A   00   00   .G.$...
>:08   00   00   00   00   00   00   00   00   ........
>:10   11   00   11   0B   11   01   11   0C   ........
>:18   11   02   11   0D   11   03   11   0E   ........
>:20   11   04   11   0F   11   05   11   10   ........
>:28   11   06   11   11   11   07   11   12   ........
>:30   11   08   11   13   11   09   11   14   ........
```

```
>:38  10 08 10 12 10 06 10 10 ........
>:40  10 04 10 0E 10 02 10 0C ........
>:48  00 00 00 00 00 00 00 00 ........
>:50  00 00 00 00 00 00 00 00 ........
>:58  00 00 00 00 00 00 00 00 ........
>:60  00 00 00 00 00 00 00 00 ........
>:68  00 00 00 00 00 00 00 00 ........
>:70  00 00 00 00 00 00 00 00 ........
>:78  00 00 00 00 00 00 00 00 ........
>:80  00 00 00 00 00 00 00 00 ........
>:88  00 00 00 00 00 00 00 00 ........
>:90  00 00 00 00 00 00 00 00 ........
>:98  00 00 00 00 00 00 00 00 ........
>:A0  00 00 00 00 00 00 00 00 ........
>:A8  00 00 00 00 00 00 00 00 ........
>:B0  00 00 00 00 00 00 00 00 ........
>:B8  00 00 00 00 00 00 00 00 ........
>:C0  00 00 00 00 00 00 00 00 ........
>:C8  00 00 00 00 00 00 00 00 ........
>:D0  00 00 00 00 00 00 00 00 ........
>:D8  00 00 00 00 00 00 00 00 ........
>:E0  00 00 00 00 00 00 00 00 ........
>:F0  00 00 00 00 00 00 00 00 ........
>:F8  00 00 00 00 00 00 00 00 ........
```

Az első két byte (a nulladik és az első) a szokásos módon a következő oldalszektor sávjára és szektorára utal (láncolás). Az általunk létrehozott file-ban a 70. rekord helyének tárolására elég egy oldalszektor, így itt a láncolási cím helyett az első oldalszektor által az ebben a blokkban lefoglalt byte-ok száma van: $47 = 71.

A második byte tartalmazza az oldalszektor sorszámát, ami nulla. Minden relatív file max. hat oldalszektort tartalmazhat és ezek számozása 0-tól indul és 5-ig tart. A harmadik byte-ban van a rekordhossz ($64 = 100). A következő 12 byte (4-től 15-ig számozott byte-ok) tartalmazza a hat oldalszektor lemezcímét (0,0 – ha a blokk még nem él).

Tehát annak ellenére, hogy az oldalszektorok össze vannak láncolva, azaz sorosan is végigolvashatjuk az összeset, a láncoláson túl még minden oldalszektor tartalmazza a saját és az összes többi lemezcímét. Egy rekord megkeresésekor a DOS a rekordhossz alapján először pontosan kiszámítja, hogy az a file hányadik adatblokkjában található. A számítás képlete a következő:

(RSZ–1)* RH/254

ahol RSZ a rekord sorszáma, RH a fix rekordhosszúság. A mi példánkban RSZ = 70, RH = 100, tehát a keresett adatblokk a 27-ik.

Az (RSZ–1)*RH megadja, hogy az RH-adik rekord a file-on belül hányadik byte-on kezdődik, és mivel a file 256 byte-os blokkokra van tördelve, ha a blokk sorszámát akarjuk megkapni, a szorzatot el kellene osztani 256-tal. Minden adatblokkban azonban az első két byte foglalt a láncolásra, ezért csak 254-gyel kell osztani, és a kapott értékhez kettőt hozzáadni. Ha a rekord (a hossztól függően) átnyúlik a következő adatblokkba, akkor két adatblokkot be kell olvasni.

Az adatblokk sorszámának ismeretében most már csak az adatblokk címére van szükség. Minden oldalszektor 120 adatblokk lemezcímét tartalmazza. A mi példánkban a 27. adatblokk lemezcíme a Ø. oldalszektorban van. Ha a számítás során 120-nál nagyobb értéket kapunk, pl. 425-öt, ezt el kell osztani 120-szal és az osztás egész része adja az oldalszektor sorszámát (3), az osztás maradéka pedig a szektoron belüli pozíciót (minden cím 2 byte-ot foglal el az oldalszektorban – alsó-, felső byte).

Foglaljuk össze a fenti, kissé bonyolultnak látszó eljárást!

A DOS lépései:

1. Kiszámítja a szükséges adatblokk sorszámát.
2. Kiszámítja, hogy az adatblokk címe hányadik oldalszektorban van.
3. Meghatározza az oldalszektor címét.
4. Beolvassa az oldalszektort, abban megkeresi az adatblokk címét.
5. Beolvassa az adatblokkot, vagy ha szükséges, a két egymást követő adatblokkot.

A fentiek alapján tehát egy rekord eléréséhez maximum négy (de lehet, hogy csak kettő) olvasási művelet elvégzésére van szükség.

```
>:00  00 F3 00 00 00 00 00 00  ........
>:08  00 00 00 00 00 00 00 00  ........
>:10  00 00 00 00 00 00 00 00  ........
>:18  00 00 00 00 00 00 00 00  ........
>:20  00 00 00 00 00 00 00 00  ........
>:28  00 00 00 00 44 41 54 45  .....70
>:30  4E 53 41 54 5A 20 37 30  REKORD
>:38  00 00 00 00 00 00 00 00  ........
>:40  00 00 00 00 00 00 00 00  ........
>:48  00 00 00 00 00 00 00 00  ........
>:50  00 00 00 00 00 00 00 00  ........
>:58  00 00 00 00 00 00 00 00  .........
>:60  00 00 00 00 00 00 00 00  ........
>:68  00 00 00 00 00 00 00 00  ........
>:70  00 00 00 00 00 00 00 00  ........
>:78  00 00 00 00 00 00 00 00  ........
>:80  00 00 00 00 00 00 00 00  ........
>:88  00 00 00 00 00 00 00 00  ........
>:90  FF 00 00 00 00 00 00 00  ........
>:98  00 00 00 00 00 00 00 00  ........
>:A0  00 00 00 00 00 00 00 00  ........
>:A8  00 00 00 00 00 00 00 00  ........
>:B0  00 00 00 00 00 00 00 00  ........
>:B8  00 00 00 00 00 00 00 00  ........
>:C0  00 00 00 00 00 00 00 00  ........
>:C8  00 00 00 00 00 00 00 00  ........
>:D0  00 00 00 00 00 00 00 00  ........
>:D8  00 00 00 00 00 00 00 00  ........
>:E0  00 00 00 00 00 00 00 00  ........
>:E8  00 00 00 00 00 00 00 00  ........
>:F0  00 00 00 00 FF 00 00 00  ........
>:F8  00 00 00 00 00 00 00 00  ........
```

Mivel a relatív file tulajdonképpeni adatblokkjai – hasonlóan a soros file-hoz – össze vannak láncolva, az adathalmazt sorosan is végigolvashatjuk vagy teleírhatjuk a relatív file szerkezeti felépítésének ismeretében.
Programozáskor még a következő elveket kell szem előtt tartani: a relatív file létrehozásához és feldolgozásához két adatcsatornát kell lefoglalni. Egyet a tényleges adatátvitelhez, egyet pedig az oldalszektorok kezeléséhez.
Amikor egy konkrét rekordra pozicionáltunk, az olvasás vagy írási művelet elvégzése előtt a DOS megvizsgálja, hogy a rekord elő van-e már készítve. Az előkészítés során az adatblokk első byte-jára $FF (CHR$ (255) karakter), a többi byte-ra $00 kerül. Ha a művelet (írás vagy olvasás) előkészítetlen blokkra vonatkozik, a DOS mindig az 50, RECORD NOT PRESENT hibaüzenetet küldi. Ez írásnál természetesen nem valódi hiba, a DOS fel is írja a rekord tartalmát a lemezre, és ezzel egyidejűleg a kisebb rekordszámú még előkészítetlen rekordokba felírja a CHR$ (255) karaktert. Előfordulhat, hogy ez hosszadalmas folyamat. Ezért célszerű a file legelső létrehozásakor az egész file-területet úgy előkészíteni, hogy lefoglaljuk a tervezett legnagyobb sorszámú rekordokat, vagyis CHR$ (255)-öt írunk bele. Ekkor az is kiderül, hogy a file elfér-e a lemezen (ha nem, azt az 52, FILE TOO LARGE hibaüzenet jelzi).
Az oldalszektorok korlátozott száma miatt egy relatív file legfeljebb 6 * 120 * 254 = 18 288 byte-ot tartalmazhat. A VC 1541-es esetében ez több, mint az egész lemezkapacitás. A nagyobb 8050-es egységnél, amely több mint 500 kbyte tárolására képes, 6 oldalszektor nem elég a lemezkapacitás kihasználásához. Ezért a 2.7 DOS változattól kezdődően kibővítették az oldalszektor eljárást (Super-Side-Sector), s itt egy relatív file maximálisan 25 Mbyte adatot tartalmazhat. Így van ez a CBM 8250-nél, a Commodore szilárdlemezeknél, valamint az újabb 8050-es egységeknél (l. még az 5.2. fejezetet).
Már említettük, hogy a relatív file két adatcsatornát igényel és mivel a VC 1541-es csak három csatorna fölött rendelkezik szabadon, ezért egyszerre csak egy relatív file lehet megnyitott állapotban. A harmadik csatornát felhasználhatjuk egy soros file egyidejű kezelésére. A nagy CBM egységeknél több csatorna áll rendelkezésre (egyidejűleg három relatív file-lal dolgozhatunk; l. az 5.2. fejezetet).

3.5. DOS lista a VC-1541-hez

```
******************************          a LED bekapcsolása
C100  78          SEI
C101  A9 F7       LDA  #F7              a LED-bit törlése
C103  2D 00 1C    AND   1C00
C106  48          PHA
C107  A5 7F       LDA   7F              egységszám
C109  F0 05       BEQ   C110            Ø ?
C10B  68          PLA
C10C  09 00       ORA  #00              a LED kikapcsolása, ha az egységszám nem Ø
C10E  D0 03       BNE   C113
C110  68          PLA
C111  09 08       ORA  #08              a LED kikapcsolása
C113  8D 00 1C    STA   1C00
C116  58          CLI
C117  60          RTS

******************************          a LED bekapcsolása
C118  78          SEI
C119  A9 08       LDA  #08
C11B  0D 00 1C    ORA   1C00            LED bekapcsolása
C11E  8D 00 1C    STA   1C00
C121  58          CLI
C122  60          RTS

******************************          kapcsolók /flag-ek törlése
C123  A9 00       LDA  #00
C125  8D 6C 02    STA   026C
C128  8D 6D 02    STA   026D
C12B  60          RTS

******************************
C12C  78          SEI
C12D  8A          TXA                   az X regiszter tartalmának mentése
C12E  48          PHA
C12F  A9 50       LDA  #50
C131  8D 6C 02    STA   026C
C134  A2 00       LDX  #00
C136  BD CA FE    LDA   FECA,X          8
C139  8D 6D 02    STA   026D
C13C  0D 00 1C    ORA   1C00
C13F  8D 00 1C    STA   1C00            LED bekapcsolása
C142  68          PLA
C143  AA          TAX                   X regiszter tartalmának betöltése
C144  58          CLI
C145  60          RTS

******************************          számítógéptől kapott utasítások kiértékelése
C146  A9 00       LDA  #00
C148  8D F9 02    STA   02F9
C14B  AD 8E 02    LDA   028E            az utoljára érintett egység száma
C14E  85 7F       STA   7F              egységszám
C150  20 BC E6    JSR   E6BC            az "ok"-üzenet előkészítése
C153  A5 84       LDA   84              másodlagos cím
C155  10 09       BPL   C160
C157  29 0F       AND  #0F
C159  C9 0F       CMP  #0F              15, utasításcsatorna
C15B  F0 03       BEQ   C160
C15D  4C B4 D7    JMP   D7B4            ugrás az OPEN-re

C160  20 B3 C2    JSR   C2B3            a sorhosszúság meghatározása, kapcsolók törlése
C163  B1 A3       LDA   (A3),Y          első karakter betöltése és tárolása
C165  8D 75 02    STA   0275
C168  A2 0B       LDX  #0B
C16A  BD 89 FE    LDA   FE89,X          a megfelelő utasítás-szó betöltése
```

```
C16D  CD 75 02   CMP   0275      az első karakter összehasonlítása
C170  F0 08      BEQ   C17A      megvan?
C172  CA         DEX
C173  10 F5      BPL   C16A
C175  A9 31      LDA   #31       nincs meg
C177  4C C8 C1   JMP   C1C8      31, "syntax error"

C17A  8E 2A 02   STX   022A      utasítás-szó sorszáma
C17D  E0 09      CPX   #09
C17F  90 03      BCC   C184      utasítás-szám < 9?
C181  20 EE C1   JSR   C1EE      "R", "S" és "N" tesztje
C184  AE 2A 02   LDX   022A      utasítás-szm
C187  BD 95 FE   LDA   FE95,X    ugrási cím LO /alsó byte/
C18A  85 6F      STA   6F
C18C  BD A1 FE   LDA   FEA1,X    ugrási cím HI /felső byte/
C18F  85 70      STA   70
C191  6C 6F 00   JMP   (006F)    ugrás az utasításra

****************************     utasítás végrehajtása után a hibaüzenet
C194  A9 00      LDA   #00       előkészítése
C196  8D F9 02   STA   02F9
C199  AD 6C 02   LDA   026C      kapcsoló be van állítva?
C19C  D0 2A      BNE   C1C8      ha igen, hibaüzenet összeállítása
C19E  A0 00      LDY   #00
C1A0  98         TYA             hiba-szám Ø
C1A1  84 80      STY   80        sáv-szám Ø
C1A3  84 81      STY   81        szektor-szám Ø
C1A5  84 A3      STY   A3
C1A7  20 C7 E6   JSR   E6C7      az "ok"-üzenet előkészítése
C1AA  20 23 C1   JSR   C123      a hibakapcsolók törlése
C1AD  A5 7F      LDA   7F        meghajtóegység-szám
C1AF  8D 8E 02   STA   028E      utolsó meghajtóegység számának feljegyzése
C1B2  AA         TAX
C1B3  A9 00      LDA   #00
C1B5  95 FF      STA   FF,X
C1B7  20 BD C1   JSR   C1BD      az input puffer törlése
C1BA  4C DA D4   JMP   D4DA      a belső csatornák lezárása

****************************     az input puffer törlése
C1BD  A0 28      LDY   #28       41 karakter törlése
C1BF  A9 00      LDA   #00
C1C1  99 00 02   STA   0200,Y    $2ØØ-től $228-ig
C1C4  88         DEY
C1C5  10 FA      BPL   C1C1
C1C7  60         RTS

****************************     a hibaüzenet /sáv + szektor/ kiírása
C1C8  A0 00      LDY   #00
C1CA  84 80      STY   80        sáv = Ø
C1CC  84 81      STY   81        szektor = Ø
C1CE  4C 45 E6   JMP   E645      hibaszám az akku-ban, ugrás a hibaüzenet
                                 létrehozására
****************************     az input-sorok ellenőrzése
C1D1  A2 00      LDX   #00
C1D3  8E 7A 02   STX   027A      az egységszám a vonatkozó mutató /pointer/
C1D6  A9 3A      LDA   #3A       ":"
C1D8  20 68 C2   JSR   C268      sor-teszt ":"-ig vagy végig
C1DB  F0 05      BEQ   C1E2      nem volt kettőspont?
C1DD  88         DEY
C1DE  88         DEY
C1DF  8C 7A 02   STY   027A      a meghajtóegység-számra mutat
C1E2  4C 68 C3   JMP   C368      az egységszám betöltése és a LED bekapcsolása

                                 input-sorok ellenőrzése
****************************
C1E5  A0 00      LDY   #00       mutató az input-pufferben
C1E7  A2 00      LDX   #00       vessző számlálása
C1E9  A9 3A      LDA   #3A       ":"
C1EB  4C 68 C2   JMP   C268      a sorok tesztelése kettőspontig vagy végig
```

```
*****************************          input sorok ellenőrzése
C1EE  20 E5 C1    JSR   C1E5           sor-teszt ":"-ig vagy végig
C1F1  D0 05       BNE   C1F8           ugrás ha kettőspont?
C1F3  A9 34       LDA   #34
C1F5  4C C8 C1    JMP   C1C8           34, "syntax error"

C1F8  88          DEY
C1F9  88          DEY                  a mutató beállítása a ":" előtti pozicióra
C1FA  8C 7A 02    STY   027A           /egységszám!/
C1FD  8A          TXA                  vessző a kettőspont előtt
C1FE  D0 F3       BNE   C1F3           ha igen, "syntax error"
C200  A9 3D       LDA   #3D            "="
C202  20 68 C2    JSR   C268           input-sor ellenőrzése
C205  8A          TXA                  vessző?
C206  F0 02       BEQ   C20A           nem
C208  A9 40       LDA   #40            a 6. a Ø. és az 5. bit beállítása
C20A  09 21       ORA   #21
C20C  8D 8B 02    STA   028B           syntaxis-ellenőrzési kapcsoló
C20F  E8          INX
C210  8E 77 02    STX   0277
C213  8E 78 02    STX   0278
C216  AD 8A 02    LDA   028A           sikerült megtalálni a jokert?
C219  F0 0D       BEQ   C228           nem
C21B  A9 80       LDA   #80
C21D  0D 8B 02    ORA   028B           a 7-es bit beállítása
C220  8D 8B 02    STA   028B
C223  A9 00       LDA   #00
C225  8D 8A 02    STA   028A           a kapcsoló visszaállítása
C228  98          TYA                  "="?
C229  F0 29       BEQ   C254           nem
C22B  9D 7A 02    STA   027A,X
C22E  AD 77 02    LDA   0277           egyenlőségjel előtti vesszők száma
C231  8D 79 02    STA   0279
C234  A9 8D       LDA   #8D            Shift + CR
C236  20 68 C2    JSR   C268           sorok ellenőrzése
C239  E8          INX                  vessző-számláló növelése
C23A  8E 78 02    STX   0278           a vesszők számának tárolása
C23D  CA          DEX
C23E  AD 8A 02    LDA   028A           sikerült megtalálni a jokert?
C241  F0 02       BEQ   C245           nem
C243  A9 08       LDA   #08            a 3-as bit beállítása
C245  EC 77 02    CPX   0277           egyenlőségjel után vessző?
C248  F0 02       BEQ   C24C           nem
C24A  09 04       ORA   #04            a 2-es bit beállítása
C24C  09 03       ORA   #03            a Ø. és az 1-es bit beállítása
C24E  4D 8B 02    EOR   028B
C251  8D 8B 02    STA   028B           szintaxis ellenőrzéshez szükséges kapcsoló
C254  AD 8B 02    LDA   028B           szintaxis kapcsoló
C257  AE 2A 02    LDX   022A           utasítás-számláló
C25A  3D A5 FE    AND   FEA5,X         összekapcsolás  az ellenőrző byte-tal
C25D  D0 01       BNE   C260           hibás szintaxis?
C25F  60          RTS

C260  8D 6C 02    STA   026C           a hiba-kapcsoló beállítása
C263  A9 30       LDA   #30
C265  4C C8 C1    JMP   C1C8           3Ø, "syntax error"

*****************************          karakter megkeresése az input-pufferben
C268  8D 75 02    STA   0275           karakter tárolása
C26B  CC 74 02    CPY   0274           már vége a sornak?
C26E  B0 2E       BCS   C29E           igen
C270  B1 A3       LDA   (A3),Y         egy karakter betöltése pufferből
C272  C8          INY
C273  CD 75 02    CMP   0275           összehasonlítása a keresett karakterrel
C276  F0 28       BEQ   C2A0           megvan
C278  C9 2A       CMP   #2A            "*"
C27A  F0 04       BEQ   C280
```

```
C27C  C9 3F     CMP  #3F       "?"
C27E  D0 03     BNE  C283
C280  EE 8A 02  INC  028A      a joker-kapcsoló beállítása
C283  C9 2C     CMP  #2C       ","
C285  D0 E4     BNE  C26B
C287  98        TYA
C288  9D 7B 02  STA  027B,X    a vessző-pozició tárolása
C28B  AD 8A 02  LDA  028A      joker-kapcsoló
C28E  29 7F     AND  #7F
C290  F0 07     BEQ  C299      nincs joker
C292  A9 80     LDA  #80
C294  95 E7     STA  E7,X      kapcsoló érték tárolása
C296  8D 8A 02  STA  028A      joker kapcsolóként
C299  E8        INX            a vessző-számláló növelése
C29A  E0 04     CPX  #04       már 4 vessző?
C29C  90 CD     BCC  C26B      nem, folytatni
C29E  A0 00     LDY  #00
C2A0  AD 74 02  LDA  0274      a sorvég kapcsoló beállítása
C2A3  9D 7B 02  STA  027B,X
C2A6  AD 8A 02  LDA  028A      joker kapcsoló
C2A9  29 7F     AND  #7F
C2AB  F0 04     BEQ  C2B1      nincs joker
C2AD  A9 80     LDA  #80
C2AF  95 E7     STA  E7,X      kapcsoló beállítása
C2B1  98        TYA
C2B2  60        RTS

******************************         sorhosszúság ellenőrzése
C2B3  A4 A3     LDY  A3        a mutató az utasításbeviteli
C2B5  F0 14     BEQ  C2CB      pufferben nulla?
C2B7  88        DEY
C2B8  F0 10     BEQ  C2CA      egyes?
C2BA  B9 00 02  LDA  0200,Y    karakter az input-pufferből
C2BD  C9 0D     CMP  #0D       "CR"
C2BF  F0 0A     BEQ  C2CB      igen, sorvége
C2C1  88        DEY
C2C2  B9 00 02  LDA  0200,Y    előtte lévő karakter
C2C5  C9 0D     CMP  #0D       "CR"
C2C7  F0 02     BEQ  C2CB      igen
C2C9  C8        INY
C2CA  C8        INY            mutató ismét a korábbi értéken
C2CB  8C 74 02  STY  0274      azonos sorhosszúság
C2CE  C0 2A     CPY  #2A       összehasonlítása 42-vel
C2D0  A0 FF     LDY  #FF
C2D2  90 08     BCC  C2DC      ha kisebb, ok
C2D4  8C 2A 02  STY  022A
C2D7  A9 32     LDA  #32
C2D9  4C C8 C1  JMP  C1C8      32,"syntax error", túl hosszú sor

******************************         az utasításbeviteli kapcsolók törlése
C2DC  A0 00     LDY  #00
C2DE  98        TYA
C2DF  85 A3     STA  A3        mutató a LO input-pufferen
C2E1  8D 58 02  STA  0258      rekordhosszúság
C2E4  8D 4A 02  STA  024A      file-tipus
C2E7  8D 96 02  STA  0296
C2EA  85 D3     STA  D3
C2EC  8D 79 02  STA  0279      vessző-számláló
C2EF  8D 77 02  STA  0277         "
C2F2  8D 78 02  STA  0278         "
C2F5  8D 8A 02  STA  028A      joker-kapcsoló
C2F8  8D 6C 02  STA  026C      hiba-kapcsoló
C2FB  A2 05     LDX  #05
C2FD  9D 79 02  STA  0279,X    soranalizís-kapcsolók
C300  95 D7     STA  D7,X      tartalomjegyzék-szektorok
C302  95 DC     STA  DC,X      puffer-mutató
C304  95 E1     STA  E1,X      meghajtóegység-számok
```

```
C306  95 E6      STA  E6,X       joker-kapcsolók
C308  9D 7F 02   STA  027F,X     sáv-számok
C30B  9D 84 02   STA  0284,X     szektor-számok
C30E  CA         DEX
C30F  D0 EC      BNE  C2FD
C311  60         RTS

****************************     egységszám átvétele
C312  AD 78 02   LDA  0278       vesszők száma
C315  8D 77 02   STA  0277       tárolás
C318  A9 01      LDA  #01
C31A  8D 78 02   STA  0278       egységszámok száma
C31D  8D 79 02   STA  0279
C320  AC 8E 02   LDY  028E       utolsó egységszám
C323  A2 00      LDX  #00
C325  86 D3      STX  D3
C327  BD 7A 02   LDA  027A,X     kettőspont poziciója
C32A  20 3C C3   JSR  C33C       kettőspont előtti egységszám beolvasása
C32D  A6 D3      LDX  D3
C32F  9D 7A 02   STA  027A,X
C332  98         TYA
C333  95 E2      STA  E2,X       egységszám a táblázatban
C335  E8         INX
C336  EC 78 02   CPX  0278       utolsó egységszám?
C339  90 EA      BCC  C325       nem, folytatás
C33B  60         RTS

****************************     egységszám megkeresése
C33C  AA         TAX             pozició tárolása
C33D  A0 00      LDY  #00
C33F  A9 3A      LDA  #3A        ":"
C341  DD 01 02   CMP  0201,X     mögötte van a kettőspont?
C344  F0 0C      BEQ  C352       igen
C346  DD 00 02   CMP  0200,X     ezen a helyen van a kettőspont?
C349  D0 16      BNE  C361       nem
C34B  E8         INX
C34C  98         TYA
C34D  29 01      AND  #01        egységszám
C34F  A8         TAY
C350  8A         TXA
C351  60         RTS

C352  BD 00 02   LDA  0200,X     az egységszám betöltése
C355  E8         INX
C356  E8         INX
C357  C9 30      CMP  #30        "Ø" ?
C359  F0 F2      BEQ  C34D       igen
C35B  C9 31      CMP  #31        "1" ?
C35D  F0 EE      BEQ  C34D       igen
C35F  D0 EB      BNE  C34C       nem, az utolsó egységszám érvényes
C361  98         TYA             utolsó egységszám
C362  09 80      ORA  #80        a 7-es bit beállítása, bizonytalan egységszám
C364  29 81      AND  #81        többi bit törlése
C366  D0 E7      BNE  C34F       az egységszám rendelkezésre bocsátása

****************************     az egységszám bevitele
C368  A9 00      LDA  #00
C36A  8D 8B 02   STA  028B       a szintaxis-kapcsoló törlése
C36D  AC 7A 02   LDY  027A       pozició az utasítás-sorban
C370  B1 A3      LDA  (A3),Y     karakter beolvasása az utasítás-pufferbe
C372  20 BD C3   JSR  C3BD       az egységszám beolvasása
C375  10 11      BPL  C388       megbízható szám?
C377  C8         INY             mutató növelése
C378  CC 74 02   CPY  0274       sor vége?
C37B  B0 06      BCS  C383       igen
C37D  AC 74 02   LDY  0274
C380  88         DEY
```

```
C381  D0 ED      BNE   C370      a meghajtó-szám utáni sor megkeresése
C383  CE 8B 02   DEC   028B
C386  A9 00      LDA   #00
C388  29 01      AND   #01
C38A  85 7F      STA   7F        meghajtó-szám
C38C  4C 00 C1   JMP   C100      a LED bekapcsolása

****************************     egységszám átkapcsolása
C38F  A5 7F      LDA   7F        egységszám
C391  49 01      EOR   #01       Ø. bit megfordítása
C393  29 01      AND   #01
C395  85 7F      STA   7F
C397  60         RTS

****************************     a file-tipus meghatározása
C398  A0 00      LDY   #00
C39A  AD 77 02   LDA   0277      sikerült megtalálni az egyenlőségjelet?
C39D  CD 78 02   CMP   0278
C3A0  F0 16      BEQ   C3B8      nem
C3A2  CE 78 02   DEC   0278      a mutató betöltése
C3A5  AC 78 02   LDY   0278
C3A8  B9 7A 02   LDA   027A,Y    mutató az "=" mögötti karakterre!
C3AB  A8         TAY
C3AC  B1 A3      LDA   (A3),Y    karakter a pufferből
C3AE  A0 04      LDY   #04       összehasonlítás a file-tipus jelöléssel
C3B0  D9 BB FE   CMP   FEBB,Y    "S", "P", "U", "R"
C3B3  F0 03      BEQ   C3B8      egyezik?
C3B5  88         DEY
C3B6  D0 F8      BNE   C3B0
C3B8  98         TYA
C3B9  8D 96 02   STA   0296      a file-tipus feljegyzése 1-től 4-ig
C3BC  60         RTS

****************************     egységszám ellenőrzése
C3BD  C9 30      CMP   #30       "Ø"
C3BF  F0 06      BEQ   C3C7
C3C1  C9 31      CMP   #31       "1"
C3C3  F0 02      BEQ   C3C7
C3C5  09 80      ORA   #80       a 7-es bit beállitása, ha nincs Ø vagy 1.
C3C7  29 81      AND   #81
C3C9  60         RTS

****************************     egységszám ellenőrzése
C3CA  A9 00      LDA   #00
C3CC  85 6F      STA   6F
C3CE  8D 8D 02   STA   028D
C3D1  48         PHA
C3D2  AE 78 02   LDX   0278      egységszámok /sorszámok/ száma
C3D5  68         PLA
C3D6  05 6F      ORA   6F
C3D8  48         PHA
C3D9  A9 01      LDA   #01
C3DB  85 6F      STA   6F
C3DD  CA         DEX
C3DE  30 0F      BMI   C3EF
C3E0  B5 E2      LDA   E2,X
C3E2  10 04      BPL   C3E8
C3E4  06 6F      ASL   6F
C3E6  06 6F      ASL   6F
C3E8  4A         LSR   A
C3E9  90 EA      BCC   C3D5
C3EB  06 6F      ASL   6F
C3ED  D0 E6      BNE   C3D5
C3EF  68         PLA
C3F0  AA         TAX
C3F1  BD 3F C4   LDA   C43F,X    a szintaxis-kapcsoló betöltése
C3F4  48         PHA
```

```
C3F5  29 03       AND  #03
C3F7  8D 8C 02    STA  028C
C3FA  68          PLA
C3FB  0A          ASL  A
C3FC  10 3E       BPL  C43C
C3FE  A5 E2       LDA  E2
C400  29 01       AND  #01     az egységszám leválasztása
C402  85 7F       STA  7F
C404  AD 8C 02    LDA  028C
C407  F0 2B       BEQ  C434
C409  20 3D C6    JSR  C63D    egység inicializálása
C40C  F0 12       BEQ  C420    nincs hiba?
C40E  20 8F C3    JSR  C38F    átkapcsolás másik egységre
C411  A9 00       LDA  #00
C413  8D 8C 02    STA  028C
C416  20 3D C6    JSR  C63D    egység inicializálása
C419  F0 1E       BEQ  C439    nincs hiba?
C41B  A9 74       LDA  #74
C41D  20 C8 C1    JSR  C1C8    74, "drive not ready"
C420  20 8F C3    JSR  C38F
C423  20 3D C6    JSR  C63D    egység inicializálása
C426  08          PHP
C427  20 8F C3    JSR  C38F    átkapcsolás másik egységre
C42A  28          PLP
C42B  F0 0C       BEQ  C439    nincs hiba?
C42D  A9 00       LDA  #00
C42F  8D 8C 02    STA  028C    egységek száma
C432  F0 05       BEQ  C439
C434  20 3D C6    JSR  C63D    egység inicializálása
C437  D0 E2       BNE  C41B    hiba?
C439  4C 00 C1    JMP  C100    LED kikapcsolása

C43C  2A          ROL  A       egységszám Carry-tól a Ø. bitig
C43D  4C 00 C4    JMP  C400

******************************    egység ellenőrzési kapcsolók
C440  00 80 41 01 01 01 01 81
C448  81 81 81 42 42 42 42

******************************    file keresése a tartalomjegyzékben
C44F  20 CA C3    JSR  C3CA    egység inicializálása
C452  A9 00       LDA  #00
C454  8D 92 02    STA  0292    mutató
C457  20 AC C5    JSR  C5AC    első tartalomjegyzék-blokk olvasása
C45A  D0 19       BNE  C475    van bejegyzés?
C45C  CE 8C 02    DEC  028C    világos az egységszám?
C45F  10 01       BPL  C462    nem
C461  60          RTS

C462  A9 01       LDA  #01
C464  8D 8D 02    STA  028D
C467  20 8F C3    JSR  C38F    átkapcsolás másik egységre
C46A  20 00 C1    JSR  C100    LED kikapcsolása
C46D  4C 52 C4    JMP  C452

C470  20 17 C6    JSR  C617    következő file keresése a tartalomjegyzékben
C473  F0 10       BEQ  C485    nem sikerült megtalálni?
C475  20 D8 C4    JSR  C4D8    bejegyzés ellenőrzése a tartalomjegyzékben
C478  AD 8F 02    LDA  028F
C47B  F0 01       BEQ  C47E    további file-ok?
C47D  60          RTS

C47E  AD 53 02    LDA  0253
C481  30 ED       BMI  C470    nem sikerült megtalálni a file-t?
C483  10 F0       BPL  C475    nem
```

```
C485  AD 8F 02   LDA  028F
C488  F0 D2      BEQ  C45C
C48A  60         RTS

C48B  20 04 C6   JSR  C604      következő tartalomjegyzék-blokk olvasása
C48E  F0 1A      BEQ  C4AA      nem sikerült megtalálni?
C490  D0 28      BNE  C4BA
C492  A9 01      LDA  #01
C494  8D 8D 02   STA  028D
C497  20 8F C3   JSR  C38F      átkapcsolás másik egységre
C49A  20 00 C1   JSR  C100      LED bekapcsolása
C49D  A9 00      LDA  #00
C49F  8D 92 02   STA  0292
C4A2  20 AC C5   JSR  C5AC      tartalomjegyzék-blokk olvasása
C4A5  D0 13      BNE  C4BA      sikerült megtalálni?
C4A7  8D 8F 02   STA  028F
C4AA  AD 8F 02   LDA  028F
C4AD  D0 28      BNE  C4D7
C4AF  CE 8C 02   DEC  028C
C4B2  10 DE      BPL  C492
C4B4  60         RTS

C4B5  20 17 C6   JSR  C617      következő bejegyzés a tartalomjegyzékben
C4B8  F0 F0      BEQ  C4AA      nem sikerült megtalálni?
C4BA  20 D8 C4   JSR  C4D8      bejegyzés ellenőrzése
C4BD  AE 53 02   LDX  0253
C4C0  10 07      BPL  C4C9      sikerült megtalálni a file-t?
C4C2  AD 8F 02   LDA  028F
C4C5  F0 EE      BEQ  C4B5      igen
C4C7  D0 0E      BNE  C4D7      ha nem, kész
C4C9  AD 96 02   LDA  0296
C4CC  F0 09      BEQ  C4D7
C4CE  B5 E7      LDA  E7,X      file-tipus
C4D0  29 07      AND  #07
C4D2  CD 96 02   CMP  0296      azonos a keresett file-tipus?
C4D5  D0 DE      BNE  C4B5      nem
C4D7  60         RTS
C4D8  A2 FF      LDX  #FF
C4DA  8E 53 02   STX  0253      megvan a file-kapcsoló
C4DD  E8         INX
C4DE  8E 8A 02   STX  028A
C4E1  20 89 C5   JSR  C589      mutató beállítása a file-re
C4E4  F0 06      BEQ  C4EC
C4E6  60         RTS

C4E7  20 94 C5   JSR  C594      mutató a következő file-on
C4EA  D0 FA      BNE  C4E6      ha vége, kész
C4EC  A5 7F      LDA  7F        egységszám
C4EE  55 E2      EOR  E2,X
C4F0  4A         LSR  A
C4F1  90 0B      BCC  C4FE
C4F3  29 40      AND  #40
C4F5  F0 F0      BEQ  C4E7
C4F7  A9 02      LDA  #02
C4F9  CD 8C 02   CMP  028C      keresés mindkét egységen
C4FC  F0 E9      BEQ  C4E7      igen
C4FE  BD 7A 02   LDA  027A,X
C501  AA         TAX
C502  20 A6 C6   JSR  C6A6      file-név hosszúság behozása
C505  A0 03      LDY  #03
C507  4C 1D C5   JMP  C51D

C50A  BD 00 02   LDA  0200,X    egy karakter betöltése az utasítássorból
C50D  D1 94      CMP  (94),Y    azonos a tartalomjegyzékben lévő karakterrel?
C50F  F0 0A      BEQ  C51B      igen
C511  C9 3F      CMP  #3F       "?"
C513  D0 D2      BNE  C4E7      nem
```

```
C515  B1 94     LDA  (94),Y
C517  C9 A0     CMP  #A0      Shift Blank, név vége?
C519  F0 CC     BEQ  C4E7     igen
C51B  E8        INX           mutató növelése
C51C  C8        INY
C51D  EC 76 02  CPX  0276     név vége az utasításban?
C520  B0 09     BCS  C52B     igen
C522  BD 00 02  LDA  0200,X   következő karakter
C525  C9 2A     CMP  #2A      "*"
C527  F0 0C     BEQ  C535     igen, sikerült megtalálni a file-t
C529  D0 DF     BNE  C50A     egyébként tovább keresni
C52B  C0 13     CPY  #13      19
C52D  B0 06     BCS  C535     név végének elérése
C52F  B1 94     LDA  (94),Y
C531  C9 A0     CMP  #A0      Shift Blank, név vége
C533  D0 B2     BNE  C4E7     nem sikerült megtalálni
C535  AE 79 02  LDX  0279
C538  8E 53 02  STX  0253
C53B  B5 E7     LDA  E7,X
C53D  29 80     AND  #80
C53F  8D 8A 02  STA  028A
C542  AD 94 02  LDA  0294
C545  95 DD     STA  DD,X
C547  A5 81     LDA  81       tartalomjegyzék szektor-száma
C549  95 D8     STA  D8,X     tárolása a táblázatban
C54B  A0 00     LDY  #00
C54D  B1 94     LDA  (94),Y   file-tipus
C54F  C8        INY
C550  48        PHA
C551  29 40     AND  #40      a scratch védelmi byte /6/ leválasztása
C553  85 6F     STA  6F       és tárolása
C555  68        PLA
C556  29 DF     AND  #DF      a 7-es bit törlése
C558  30 02     BMI  C55C
C55A  09 20     ORA  #20      az 5-ös bit beállítása
C55C  29 27     AND  #27      a 3-as és 4-es bit törlése
C55E  05 6F     ORA  6F       a 6-os bit ismétlése
C560  85 6F     STA  6F
C562  A9 80     LDA  #80
C564  35 E7     AND  E7,X     a joker kapcsoló leválasztása
C566  05 6F     ORA  6F
C568  95 E7     STA  E7,X     joker kapcsoló a táblázatba
C56A  B5 E2     LDA  E2,X
C56C  29 80     AND  #80
C56E  05 7F     ORA  7F       egységszám
C570  95 E2     STA  E2,X
C572  B1 94     LDA  (94),Y
C574  9D 80 02  STA  0280,X   első file-sáv
C577  C8        INY
C578  B1 94     LDA  (94),Y
C57A  9D 85 02  STA  0285,X   és a szektor beolvasása a tartalomjegyzékből
C57D  AD 58 02  LDA  0258     rekordhosszúság
C580  D0 07     BNE  C589     ellenőrzés
C582  A0 15     LDY  #15
C584  B1 94     LDA  (94),Y   rekordhosszúság
C586  8D 58 02  STA  0258     beolvasása a tartalomjegyzékből
C589  A9 FF     LDA  #FF
C58B  8D 8F 02  STA  028F
C58E  AD 78 02  LDA  0278
C591  8D 79 02  STA  0279
C594  CE 79 02  DEC  0279
C597  10 01     BPL  C59A
C599  60        RTS

C59A  AE 79 02  LDX  0279
C59D  B5 E7     LDA  E7,X     joker kapcsoló magas?
C59F  30 05     BMI  C5A6     igen
```

```
C5A1  BD 80 02   LDA  0280,X     be van már állítva a sáv-szám?
C5A4  D0 EE      BNE  C594       igen
C5A6  A9 00      LDA  #00
C5A8  8D 8F 02   STA  028F
C5AB  60         RTS

C5AC  A0 00      LDY  #00
C5AE  8C 91 02   STY  0291
C5B1  88         DEY
C5B2  8C 53 02   STY  0253
C5B5  AD 85 FE   LDA  FE85       18, tartalomjegyzék sávja
C5B8  85 80      STA  80
C5BA  A9 01      LDA  #01
C5BC  85 81      STA  81         1. szektor
C5BE  8D 93 02   STA  0293
C5C1  20 75 D4   JSR  D475       szektor olvasása
C5C4  AD 93 02   LDA  0293
C5C7  D0 01      BNE  C5CA
C5C9  60         RTS

C5CA  A9 07      LDA  #07
C5CC  8D 95 02   STA  0295       file bejegyzések száma /-1/
C5CF  A9 00      LDA  #00
C5D1  20 F6 D4   JSR  D4F6       egy karakter beolvasása a pufferból
C5D4  8D 93 02   STA  0293       tárolása /sáv-szám/
C5D7  20 E8 D4   JSR  D4E8       a puffer mutató beállítása
C5DA  CE 95 02   DEC  0295       a számláló csökkentése
C5DD  A0 00      LDY  #00
C5DF  B1 94      LDA  (94),Y     a tartalomjegyzék első byte-ja
C5E1  D0 18      BNE  C5FB
C5E3  AD 91 02   LDA  0291
C5E6  D0 2F      BNE  C617
C5E8  20 3B DE   JSR  DE3B       a sáv és a szektor-szám betöltése
C5EB  A5 81      LDA  81
C5ED  8D 91 02   STA  0291       szektor-szám
C5F0  A5 94      LDA  94
C5F2  AE 92 02   LDX  0292
C5F5  8D 92 02   STA  0292       puffer mutató
C5F8  F0 1D      BEQ  C617
C5FA  60         RTS

C5FB  A2 01      LDX  #01
C5FD  EC 92 02   CPX  0292       puffer mutató egyesen?
C600  D0 2D      BNE  C62F
C602  F0 13      BEQ  C617
C604  AD 85 FE   LDA  FE85       18, BAM sáv száma
C607  85 80      STA  80         sáv-szám
C609  AD 90 02   LDA  0290
C60C  85 81      STA  81         szektor-szám
C60E  20 75 D4   JSR  D475       ugrás egy blokk olvasására
C611  AD 94 02   LDA  0294
C614  20 C8 D4   JSR  D4C8       a puffer mutató beállítása
C617  A9 FF      LDA  #FF
C619  8D 53 02   STA  0253       megtalált file-kapcsoló törlése
C61C  AD 95 02   LDA  0295
C61F  30 08      BMI  C629       megtörtént az összes file bejegyzés ellen-
C621  A9 20      LDA  #20        őrzése?
C623  20 C6 D1   JSR  D1C6       puffer mutató növelése 32-vel
C626  4C D7 C5   JMP  C5D7       következő bejegyzés és további keresés

C629  20 4D D4   JSR  D44D       puffer mutató beállítása
C62C  4C C4 C5   JMP  C5C4       a következő blokk olvasása

C62F  A5 94      LDA  94
C631  8D 94 02   STA  0294
C634  20 3B DE   JSR  DE3B       sáv és szektor-szám beolvasása a pufferból
```

```
C637  A5 81      LDA   81
C639  8D 90 02   STA   0290      szektor-szám tárolása
C63C  60         RTS

****************************     meghajtóegység tesztelése és inicializálása
C63D  A5 68      LDA   68
C63F  D0 28      BNE   C669
C641  A6 7F      LDX   7F        egységszám
C643  56 1C      LSR   1C,X      volt lemezcsere?
C645  90 22      BCC   C669      ha nem, kész
C647  A9 FF      LDA   #FF
C649  8D 98 02   STA   0298      a hiba-kapcsoló beállítása
C64C  20 0E D0   JSR   D00E      a tartalomjegyzék-sáv olvasása
C64F  A0 FF      LDY   #FF
C651  C9 02      CMP   #02       20,"read error"?
C653  F0 0A      BEQ   C65F      igen
C655  C9 03      CMP   #03       21,"read error"?
C657  F0 06      BEQ   C65F      igen
C659  C9 0F      CMP   #0F       74, "drive not ready"?
C65B  F0 02      BEQ   C65F      igen
C65D  A0 00      LDY   #00
C65F  A6 7F      LDX   7F        egységszám
C661  98         TYA
C662  95 FF      STA   FF,X      a hiba-kapcsoló tárolása
C664  D0 03      BNE   C669      hiba?
C666  20 42 D0   JSR   D042      BAM betöltése
C669  A6 7F      LDX   7F        egységszám
C66B  B5 FF      LDA   FF,X      a hibakód átadása
C66D  60         RTS

****************************     file neve a tartalomjegyzék pufferben
C66E  48         PHA
C66F  20 A6 C6   JSR   C6A6
C672  20 88 C6   JSR   C688      a file-név betöltése a pufferbe
C675  68         PLA
C676  38         SEC
C677  ED 4B 02   SBC   024B      hosszúság összehasonlítása a maximális
C67A  AA         TAX             hosszal
C67B  F0 0A      BEQ   C687
C67D  90 08      BCC   C687
C67F  A9 A0      LDA   #A0       "Shift Space"-szel kitölteni
C681  91 94      STA   (94),Y
C683  C8         INY
C684  CA         DEX
C685  D0 FA      BNE   C681
C687  60         RTS

****************************
C688  98         TYA             puffer-szám
C689  0A         ASL   A
C68A  A8         TAY             kétszerese a mutatónak
C68B  B9 99 00   LDA   0099,Y
C68E  85 94      STA   94
C690  B9 9A 00   LDA   009A,Y    karakter átvitele a pufferbe
C693  85 95      STA   95
C695  A0 00      LDY   #00
C697  BD 00 02   LDA   0200,X
C69A  91 94      STA   (94),Y
C69C  C8         INY
C69D  F0 06      BEQ   C6A5      megtelt a puffer?
C69F  E8         INX
C6A0  EC 76 02   CPX   0276
C6A3  90 F2      BCC   C697
C6A5  60         RTS
```

```
******************************  név végének megkeresése az utasítástárban
C6A6  A9 00     LDA  #00
C6A8  8D 4B 02  STA  024B        hosszúság előfoglalás
C6AB  8A        TXA
C6AC  48        PHA
C6AD  BD 00 02  LDA  0200,X      karakter beolvasása a pufferből
C6B0  C9 2C     CMP  #2C         ","
C6B2  F0 14     BEQ  C6C8
C6B4  C9 3D     CMP  #3D         "="
C6B6  F0 10     BEQ  C6C8
C6B8  EE 4B 02  INC  024B        a név hosszának növelése
C6BB  E8        INX
C6BC  A9 0F     LDA  #0F         15
C6BE  CD 4B 02  CMP  024B
C6C1  90 05     BCC  C6C8        nagyobb?
C6C3  EC 74 02  CPX  0274        vége az input-sornak?
C6C6  90 E5     BCC  C6AD
C6C8  8E 76 02  STX  0276
C6CB  68        PLA
C6CC  AA        TAX              mutató a név végén
C6CD  60        RTS

******************************
C6CE  A5 83     LDA  83
C6D0  48        PHA              másodlagos cím és csatorna-szám
C6D1  A5 82     LDA  82
C6D3  48        PHA
C6D4  20 DE C6  JSR  C6DE        file bejegyzés előállítása
C6D7  68        PLA
C6D8  85 82     STA  82
C6DA  68        PLA              az adatok visszatöltése
C6DB  85 83     STA  83
C6DD  60        RTS

******************************
C6DE  A9 11     LDA  #11         17
C6E0  85 83     STA  83          másodlagos cím
C6E2  20 EB D0  JSR  D0EB        olvasó csatorna megnyitása
C6E5  20 E8 D4  JSR  D4E8        a puffer mutató beállítása
C6E8  AD 53 02  LDA  0253
C6EB  10 0A     BPL  C6F7        még nem az utolsó bejegyzés?
C6ED  AD 8D 02  LDA  028D
C6F0  D0 0A     BNE  C6FC
C6F2  20 06 C8  JSR  C806        "blocks free." írása
C6F5  18        CLC
C6F6  60        RTS

C6F7  AD 8D 02  LDA  028D
C6FA  F0 1F     BEQ  C71B
C6FC  CE 8D 02  DEC  028D
C6FF  D0 0D     BNE  C70E
C701  CE 8D 02  DEC  028D
C704  20 8F C3  JSR  C38F        áttérés másik egységre
C707  20 06 C8  JSR  C806        "blocks free." írása
C70A  38        SEC
C70B  4C 8F C3  JMP  C38F        áttérés másik egységre

C70E  A9 00     LDA  #00
C710  8D 73 02  STA  0273        egységszám felülíráshoz, felső byte
C713  8D 8D 02  STA  028D
C716  20 B7 C7  JSR  C7B7        felülírás
C719  38        SEC
C71A  60        RTS

C71B  A2 18     LDX  #18
C71D  A0 1D     LDY  #1D
C71F  B1 94     LDA  (94),Y      hi blokkok száma
```

```
C721  8D 73 02   STA  0273      a pufferben
C724  F0 02      BEQ  C728      nulla?
C726  A2 16      LDX  #16
C728  88         DEY
C729  B1 94      LDA  (94),Y    lo blokkok száma
C72B  8D 72 02   STA  0272      a pufferben
C72E  E0 16      CPX  #16
C730  F0 0A      BEQ  C73C
C732  C9 0A      CMP  #0A       10
C734  90 06      BCC  C73C
C736  CA         DEX
C737  C9 64      CMP  #64       100
C739  90 01      BCC  C73C
C73B  CA         DEX
C73C  20 AC C7   JSR  C7AC      puffer törlése
C73F  B1 94      LDA  (94),Y    file-tipus
C741  48         PHA
C742  0A         ASL  A         7-es bit a carry-ba
C743  10 05      BPL  C74A      nincs beállítva a 6-os bit?
C745  A9 3C      LDA  #3C       "<"-t a védett file-hoz
C747  9D B2 02   STA  02B2,X    a file-tipus után beírni
C74A  68         PLA
C74B  29 0F      AND  #0F       0-3 bit leválasztása
C74D  A8         TAY            file-tipus jelölési indexként
C74E  B9 C5 FE   LDA  FEC5,Y    file-tipus 3. betűje
C751  9D B1 02   STA  02B1,X    a pufferben
C754  CA         DEX
C755  B9 C0 FE   LDA  FEC0,Y    file-tipus 2. betűje
C758  9D B1 02   STA  02B1,X    a pufferben
C75B  CA         DEX
C75C  B9 BB FE   LDA  FEBB,Y    file-tipus 1. betűje
C75F  9D B1 02   STA  02B1,X    a pufferben
C762  CA         DEX
C763  CA         DEX
C764  B0 05      BCS  C76B      nincs lezárva a file?
C766  A9 2A      LDA  #2A       "*"
C768  9D B2 02   STA  02B2,X    file-tipus előtti helyek kitöltése
C76B  A9 A0      LDA  #A0       "shift space"-szel
C76D  9D B1 02   STA  02B1,X
C770  CA         DEX
C771  A0 12      LDY  #12
C773  B1 94      LDA  (94),Y    file-név
C775  9D B1 02   STA  02B1,X    beírása a pufferbe
C778  CA         DEX
C779  88         DEY
C77A  C0 03      CPY  #03
C77C  B0 F5      BCS  C773
C77E  A9 22      LDA  #22       '"'
C780  9D B1 02   STA  02B1,X    file-név előtt beírni
C783  E8         INX
C784  E0 20      CPX  #20
C786  B0 0B      BCS  C793
C788  BD B1 02   LDA  02B1,X    karakter a pufferből
C78B  C9 22      CMP  #22       '"'?
C78D  F0 04      BEQ  C793
C78F  C9 A0      CMP  #A0       a név végén a "shift space"
C791  D0 F0      BNE  C783
C793  A9 22      LDA  #22       '"'-el helyettesíteni
C795  9D B1 02   STA  02B1,X
C798  E8         INX
C799  E0 20      CPX  #20
C79B  B0 0A      BCS  C7A7
C79D  A9 7F      LDA  #7F       7-es bit
C79F  3D B1 02   AND  02B1,X
C7A2  9D B1 02   STA  02B1,X
C7A5  10 F1      BPL  C798
```

```
C7A7   20 B5 C4    JSR   C4B5      áttérés a következő tartalomjegyzék bejegy-
C7AA   38          SEC             zésre
C7AB   60          RTS

*****************************      a puffer törlése
C7AC   A0 1B       LDY   #1B
C7AE   A9 20       LDA   #20       ' ' Blank-t
C7B0   99 B0 02    STA   02B0,Y    a pufferbe beírni
C7B3   88          DEY
C7B4   D0 FA       BNE   C7B0
C7B6   60          RTS

*****************************
C7B7   20 19 F1    JSR   F119      lemez inicializálása, ha lemezcsere volt
C7BA   20 DF F0    JSR   F0DF      lemez nevének olvasása
C7BD   20 AC C7    JSR   C7AC      puffer törlése
C7C0   A9 FF       LDA   #FF
C7C2   85 6F       STA   6F
C7C4   A6 7F       LDX   7F        egységszám
C7C6   8E 72 02    STX   0272      blokkszám alsó byte
C7C9   A9 00       LDA   #00
C7CB   8D 73 02    STA   0273      blokkok száma /Lo/
C7CE   A6 F9       LDX   F9        puffer-szám
C7D0   BD E0 FE    LDA   FEE0,X    puffer-cím felső byte-ja /Hi/
C7D3   85 95       STA   95
C7D5   AD 88 FE    LDA   FE88      $90, lemeznév poziciójának
C7D8   85 94       STA   94        tárolása
C7DA   A0 16       LDY   #16
C7DC   B1 94       LDA   (94),Y    "shift space"-kal a puffer kitöltése
C7DE   C9 A0       CMP   #A0
C7E0   D0 0B       BNE   C7ED
C7E2   A9 31       LDA   #31       "1"
C7E4   2C B1 94    BIT   94B1      pufferből származó karakter
C7E7   C9 A0       CMP   #A0       összehasonlítása a "shift space"-vel
C7E9   D0 02       BNE   C7ED
C7EB   A9 20       LDA   #20       ' ' space
C7ED   99 B3 02    STA   02B3,Y    a pufferben
C7F0   88          DEY
C7F1   10 F2       BPL   C7E5
C7F3   A9 12       LDA   #12       "RVS ON"
C7F5   8D B1 02    STA   02B1      a pufferben
C7F8   A9 22       LDA   #22       '"' -t
C7FA   8D B2 02    STA   02B2      lemeznév elé
C7FD   8D C3 02    STA   02C3      és utána beírni
C800   A9 20       LDA   #20       ' ' space
C802   8D C4 02    STA   02C4      mögötte
C805   60          RTS

*****************************      a lezáró sor előállítása
C806   20 AC C7    JSR   C7AC      puffer törlése
C809   A0 0B       LDY   #0B       12 "blocks free"
C80B   B9 17 C8    LDA   C817,Y    jel
C80E   99 B1 02    STA   02B1,Y    beírása a pufferbe
C811   88          DEY
C812   10 F7       BPL   C80B
C814   4C 4D EF    JMP   EF4D      szabad blokkok száma

*****************************
C817   42 4C 4F 43 4B 53 20 46     "blocks free"
C81F   52 45 45 2E

*****************************      "Scratch" S-utasítás
C823   20 98 C3    JSR   C398      a file-tipus meghatározása
C826   20 20 C3    JSR   C320      egységszám betöltése
C829   20 CA C3    JSR   C3CA      szükséges esetben az egység inicializálása
C82C   A9 00       LDA   #00
C82E   85 86       STA   86        törölt file-ok
```

```
C830  20 9D C4   JSR  C49D     file megkeresése a tartalomjegyzékben
C833  30 3D      BMI  C872     nem sikerült megtalálni?
C835  20 B7 DD   JSR  DDB7     megnyitott állapotban van a file?
C838  90 33      BCC  C86D     igen
C83A  A0 00      LDY  #00
C83C  B1 94      LDA  (94),Y   file-tipus
C83E  29 40      AND  #40      védelem?
C840  D0 2B      BNE  C86D     igen
C842  20 B6 C8   JSR  C8B6     file törlése
C845  A0 13      LDY  #13
C847  B1 94      LDA  (94),Y   első oldal-szektor sáv-száma
C849  F0 0A      BEQ  C855     nincs egy sem?
C84B  85 80      STA  80       sáv-szám feljegyzése
C84D  C8         INY
C84E  B1 94      LDA  (94),Y   és a szektor-szám
C850  85 81      STA  81
C852  20 7D C8   JSR  C87D     oldal-szektorok törlés
C855  AE 53 02   LDX  0253     file-szám
C858  A9 20      LDA  #20
C85A  35 E7      AND  E7,X     az 5-ös bit magas?
C85C  D0 0D      BNE  C86B     igen, a file nincs lezárva
C85E  BD 80 02   LDA  0280,X   sáv
C861  85 80      STA  80
C863  BD 85 02   LDA  0285,X   és szektor betöltése
C866  85 81      STA  81
C868  20 7D C8   JSR  C87D     file törlése
C86B  E6 86      INC  86       törölt file-ok számának növelése
C86D  20 8B C4   JSR  C48B     áttérés a következő file-ra
C870  10 C3      BPL  C835     törlés
C872  A5 86      LDA  86       törölt file-ok száma
C874  85 80      STA  80       tárolása
C876  A9 01      LDA  #01      1, mint disk-státus
C878  A0 00      LDY  #00      Ø, mint szektor
C87A  4C A3 C1   JMP  C1A3     a "files scratched" üzenet előkészítése

*****************************  egy file törlése
C87D  20 5F EF   JSR  EF5F     a blokkok felszabadítása a BAM-ban
C880  20 75 D4   JSR  D475
C883  20 19 F1   JSR  F119     BAM puffer számnak a betöltése
C886  B5 A7      LDA  A7,X
C888  C9 FF      CMP  #FF
C88A  F0 08      BEQ  C894
C88C  AD F9 02   LDA  02F9
C88F  09 40      ORA  #40
C891  8D F9 02   STA  02F9
C894  A9 00      LDA  #00
C896  20 C8 D4   JSR  D4C8     puffer-mutató nullán
C899  20 56 D1   JSR  D156     a sáv-szám betöltése
C89C  85 80      STA  80
C89E  20 56 D1   JSR  D156     a szektor-szám betöltése
C8A1  85 81      STA  81
C8A3  A5 80      LDA  80       sáv-szám
C8A5  D0 06      BNE  C8AD     nem egyenlő nullával?
C8A7  20 F4 EE   JSR  EEF4     BAM-bejegyzések
C8AA  4C 27 D2   JMP  D227     a csatorna lezárása

C8AD  20 5F EF   JSR  EF5F     a blokk felszabadítása a BAM-ban
C8B0  20 4D D4   JSR  D44D     a következő blokk olvasása
C8B3  4C 94 C8   JMP  C894     és folytatása

*****************************  a file-bejegyzés törlése
C8B6  A0 00      LDY  #00
C8B8  98         TYA
C8B9  91 94      STA  (94),Y   file-tipus nullára állítása
C8BB  20 5E DE   JSR  DE5E     blokk írása
C8BE  4C 99 D5   JMP  D599     és ellenőrzése
```

```
****************************       D-utasítás, "Backup"
C8C1  A9 31      LDA  #31
C8C3  4C C8 C1   JMP   C1C8        31, "syntax error"
                                   lemez formálása
C8C6  A9 4C      LDA  #4C          JMP-utasítás
C8C8  8D 00 06   STA   0600
C8CB  A9 C7      LDA  #C7
C8CD  8D 01 06   STA   0601        JMP $FAC7 a $600-$602 után
C8D0  A9 FA      LDA  #FA
C8D2  8D 02 06   STA   0602
C8D5  A9 03      LDA  #03
C8D7  20 D3 D6   JSR   D6D3        a sáv- és a szektor-szám beállítása
C8DA  A5 7F      LDA   7F          egységszám
C8DC  09 E0      ORA  #E0          a formálás utasításkódjának
C8DE  85 03      STA   03          átadása
C8E0  A5 03      LDA   03
C8E2  30 FC      BMI   C8E0        a formálás végén
C8E4  C9 02      CMP  #02          visszajelzés ellenőrzése
C8E6  90 07      BCC   C8EF        ha 2-nél kisebb, ok
C8E8  A9 03      LDA  #03
C8EA  A2 00      LDX  #00
C8EC  4C 0A E6   JMP   E60A        21, "read error"

C8EF  60         RTS

****************************       C-utasítás, "Copy"
C8F0  A9 E0      LDA  #E0
C8F2  8D 4F 02   STA   024F
C8F5  20 D1 F0   JSR   F0D1
C8F8  20 19 F1   JSR   F119        BAM puffer számának betöltése
C8FB  A9 FF      LDA  #FF
C8FD  95 A7      STA   A7,X
C8FF  A9 0F      LDA  #0F
C901  8D 56 02   STA   0256
C904  20 E5 C1   JSR   C1E5        az input-sor ellenőrzése
C907  D0 03      BNE   C90C
C909  4C C1 C8   JMP   C8C1        31, "syntax error"

C90C  20 F8 C1   JSR   C1F8        az input-sor ellenőrzése
C90F  20 20 C3   JSR   C320        meghajtószámok tesztelése
C912  AD 8B 02   LDA   028B        szintaxis-ellenőrzés kapcsoló
C915  29 55      AND  #55
C917  D0 0F      BNE   C928
C919  AE 7A 02   LDX   027A
C91C  BD 00 02   LDA   0200,X      az utasítás karaktere
C91F  C9 2A      CMP  #2A          "*"
C921  D0 05      BNE   C928
C923  A9 30      LDA  #30
C925  4C C8 C1   JMP   C1C8        30,"syntax error"

C928  AD 8B 02   LDA   028B        szintaxis-kapcsoló
C92B  29 D9      AND  #D9
C92D  D0 F4      BNE   C923        30, "syntax error"
C92F  4C 52 C9   JMP   C952

C932  A9 00      LDA  #00
C934  8D 58 02   STA   0258
C937  8D 8C 02   STA   028C        egységek száma
C93A  8D 80 02   STA   0280        sáv-szám a tartalomjegyzékben
C93D  8D 81 02   STA   0281
C940  A5 E3      LDA   E3
C942  29 01      AND  #01
C944  85 7F      STA   7F          egységszám
C946  09 01      ORA  #01
C948  8D 91 02   STA   0291
```

```
C94B  AD 7B 02   LDA  027B
C94E  8D 7A 02   STA  027A
C951  60         RTS

C952  20 4F C4   JSR  C44F   file megkeresése a tartalomjegyzékben
C955  AD 7B 02   LDA  027B   file-nevek száma az utasításban
C958  C9 03      CMP  #03    3-nál kisebb?
C95A  90 45      BCC  C9A1   igen
C95C  A5 E2      LDA  E2     első egységszám
C95E  C5 E3      CMP  E3     második egységszám
C960  D0 3F      BNE  C9A1   nem azonos egység?
C962  A5 DD      LDA  DD     első file tartalomjegyzék-blokkja
C964  C5 DE      CMP  DE     azonos a 2. file tartalomjegyzék-blokkjával?
C966  D0 39      BNE  C9A1   nem
C968  A5 D8      LDA  D8     első file tartalomjegyzék-szektora
C96A  C5 D9      CMP  D9     azonos a 2. file tartalomjegyzék-szektorával?
C96C  D0 33      BNE  C9A1   nem
C96E  20 CC CA   JSR  CACC   megvan a file?
C971  A9 01      LDA  #01
C973  8D 79 02   STA  0279
C976  20 FA C9   JSR  C9FA
C979  20 25 D1   JSR  D125   a file-tipus beolvasása
C97C  F0 04      BEQ  C982   relatív file?
C97E  C9 02      CMP  #02    prg-file
C980  D0 05      BNE  C987   nem
C982  A9 64      LDA  #64
C984  20 C8 C1   JSR  C1C8   64, "file type mismatch"
C987  A9 12      LDA  #12    18
C989  85 83      STA  83     másodlagos cím
C98B  AD 3C 02   LDA  023C
C98E  8D 3D 02   STA  023D
C991  A9 FF      LDA  #FF
C993  8D 3C 02   STA  023C
C996  20 2A DA   JSR  DA2A   append előkészítése
C999  A2 02      LDX  #02
C99B  20 B9 C9   JSR  C9B9   file-ok másolása
C99E  4C 94 C1   JMP  C194   kész

C9A1  20 A7 C9   JSR  C9A7   file-ok másolása
C9A4  4C 94 C1   JMP  C194   kész

C9A7  20 E7 CA   JSR  CAE7
C9AA  A5 E2      LDA  E2     első file egységszáma
C9AC  29 01      AND  #01
C9AE  85 7F      STA  7F     egységszám
C9B0  20 86 D4   JSR  D486
C9B3  20 E4 D6   JSR  D6E4   a file bejegyzése a tartalomjegyzékbe
C9B6  AE 77 02   LDX  0277
C9B9  8E 79 02   STX  0279
C9BC  20 FA C9   JSR  C9FA
C9BF  A9 11      LDA  #11    17
C9C1  85 83      STA  83
C9C3  20 EB D0   JSR  D0EB
C9C6  20 25 D1   JSR  D125   a file-tipus betöltése
C9C9  D0 03      BNE  C9CE   nem relatív file?
C9CB  20 53 CA   JSR  CA53
C9CE  A9 08      LDA  #08
C9D0  85 F8      STA  F8
C9D2  4C D8 C9   JMP  C9D8

C9D5  20 9B CF   JSR  CF9B   byte beírása a pufferbe
C9D8  20 35 CA   JSR  CA35   és betöltése
C9DB  A9 80      LDA  #80
C9DD  20 A6 DD   JSR  DDA6   a 7-es bit tesztje
C9E0  F0 F3      BEQ  C9D5   nincs beállítva?
C9E2  20 25 D1   JSR  D125   file-tipus ellenőrzése
C9E5  F0 03      BEQ  C9EA   relatív file?
```

```
C9E7  20 9B CF   JSR  CF9B    adat-byte beolvasása a pufferbe
C9EA  AE 79 02   LDX  0279
C9ED  E8         INX
C9EE  EC 78 02   CPX  027B
C9F1  90 C6      BCC  C9B9
C9F3  A9 12      LDA  #12     18
C9F5  85 83      STA  83
C9F7  4C 02 DB   JMP  DB02    a csatorna lezárása

C9FA  AE 79 02   LDX  0279
C9FD  B5 E2      LDA  E2,X    egységszám
C9FF  29 01      AND  #01
CA01  85 7F      STA  7F      tárolása
CA03  AD 85 FE   LDA  FE85    18, tartalomjegyzék-sáv
CA06  85 80      STA  80      tárolása
CA08  B5 D8      LDA  D8,X    tartalomjegyzék-szektor
CA0A  85 81      STA  81
CA0C  20 75 D4   JSR  D475    blokk olvasása
CA0F  AE 79 02   LDX  0279
CA12  B5 DD      LDA  DD,X    mutató a blokkban
CA14  20 C8 D4   JSR  D4C8    puffer mutató beállítása
CA17  AE 79 02   LDX  0279
CA1A  B5 E7      LDA  E7,X    file-tipus
CA1C  29 07      AND  #07     leválasztása
CA1E  8D 4A 02   STA  024A    és tárolása
CA21  A9 00      LDA  #00
CA23  8D 58 02   STA  0258
CA26  20 A0 D9   JSR  D9A0    relatív file paramétereinek olvasása
CA29  A0 01      LDY  #01
CA2B  20 25 D1   JSR  D125    file-tipus beolvasása
CA2E  F0 01      BEQ  CA31    relatív file?
CA30  C8         INY
CA31  98         TYA
CA32  4C C8 D4   JMP  D4C8    a puffer mutató beállítása

CA35  A9 11      LDA  #11     17
CA37  85 83      STA  83
CA39  20 9B D3   JSR  D39B    a csatorna megnyitása és a byte betöltése
CA3C  85 85      STA  85
CA3E  A6 82      LDX  82      csatorna-szám
CA40  B5 F2      LDA  F2,X
CA42  29 08      AND  #08     a végjelölések leválasztása
CA44  85 F8      STA  F8
CA46  D0 0A      BNE  CA52    nincs beállítva?
CA48  20 25 D1   JSR  D125    file-tipus betöltése
CA4B  F0 05      BEQ  CA52    relatív file?
CA4D  A9 80      LDA  #80
CA4F  20 97 DD   JSR  DD97    a 7-es bit beállítása
CA52  60         RTS

CA53  20 D3 D1   JSR  D1D3    az egységszám beállítása
CA56  20 CB E1   JSR  E1CB
CA59  A5 D6      LDA  D6
CA5B  48         PHA
CA5C  A5 D5      LDA  D5
CA5E  48         PHA
CA5F  A9 12      LDA  #12     18
CA61  85 83      STA  83
CA63  20 07 D1   JSR  D107    írócsatorna megnyitása
CA66  20 D3 D1   JSR  D1D3    egységszám beállítása
CA69  20 CB E1   JSR  E1CB
CA6C  20 9C E2   JSR  E29C
CA6F  A5 D6      LDA  D6
CA71  85 87      STA  87
CA73  A5 D5      LDA  D5
CA75  85 86      STA  86
CA77  A9 00      LDA  #00
```

```
CA79  85 88     STA  88
CA7B  85 D4     STA  D4
CA7D  85 D7     STA  D7
CA7F  68        PLA
CA80  85 D5     STA  D5
CA82  68        PLA
CA83  85 D6     STA  D6
CA85  4C 3B E3  JMP  E33B

****************************        R-utasítás "Rename"
CA88  20 20 C3  JSR  C320           egységszám betöltése az utasítás-sorból
CA8B  A5 E3     LDA  E3
CA8D  29 01     AND  #01
CA8F  85 E3     STA  E3             2. egységszám
CA91  C5 E2     CMP  E2             összehasonlítása az első egységszámmal
CA93  F0 02     BEQ  CA97           azonos?
CA95  09 80     ORA  #80
CA97  85 E2     STA  E2
CA99  20 4F C4  JSR  C44F           file megkeresése a tartalomjegyzékben
CA9C  20 E7 CA  JSR  CAE7           léteznek a nevek?
CA9F  A5 E3     LDA  E3
CAA1  29 01     AND  #01
CAA3  85 7F     STA  7F             egységszám
CAA5  A5 D9     LDA  D9
CAA7  85 81     STA  81             szektor-szám
CAA9  20 57 DE  JSR  DE57           a blokk beolvasása a tartalomjegyzékből
CAAC  20 99 D5  JSR  D599           O K -t olvas?
CAAF  A5 DE     LDA  DE             a tartalomjegyzék bejegyzés mutatója
CAB1  18        CLC
CAB2  69 03     ADC  #03
CAB4  20 C8 D4  JSR  D4C8           a puffer mutató beállítása
CAB7  20 93 DF  JSR  DF93           a puffer számának betöltése
CABA  A8        TAY
CABB  AE 7A 02  LDX  027A
CABE  A9 10     LDA  #10            16 karakter
CAC0  20 6E C6  JSR  C66E           név beírása a pufferbe
CAC3  20 5E DE  JSR  DE5E           blokkok felírása a lemezre
CAC6  20 99 D5  JSR  D599           Ok?
CAC9  4C 94 C1  JMP  C194           kész, lemez-státus előkészítése

****************************        file-ok lemezen való jelenlétének ellenőrzése
CACC  A5 E8     LDA  E8             file-tipus
CACE  29 07     AND  #07
CAD0  8D 4A 02  STA  024A           tárolása
CAD3  AE 78 02  LDX  0278
CAD6  CA        DEX
CAD7  EC 77 02  CPX  0277
CADA  90 0A     BCC  CAE6
CADC  BD 80 02  LDA  0280,X         sáv-szám
CADF  D0 F5     BNE  CAD6           nem egyenlő nullával?
CAE1  A9 62     LDA  #62
CAE3  4C C8 C1  JMP  C1C8           62, "file not found"

CAE6  60        RTS

CAE7  20 CC CA  JSR  CACC           van korábbi nevű file?
CAEA  BD 80 02  LDA  0280,X         új file-ok sáv-száma
CAED  F0 05     BEQ  CAF4           volt file-törlés?
CAEF  A9 63     LDA  #63
CAF1  4C C8 C1  JMP  C1C8           63, "file exists"

CAF4  CA        DEX
CAF5  10 F3     BPL  CAEA
CAF7  60        RTS
```

```
******************************        M-utasítás, "Memory"
CAF8  AD 01 02   LDA   0201           a második karakter a pufferből
CAFB  C9 2D      CMP   #2D            "-"
CAFD  D0 4C      BNE   CB4B
CAFF  AD 03 02   LDA   0203
CB02  85 6F      STA   6F             cím a $6F/$70 után
CB04  AD 04 02   LDA   0204
CB07  85 70      STA   70
CB09  A0 00      LDY   #00
CB0B  AD 02 02   LDA   0202           3. karakter a pufferből
CB0E  C9 52      CMP   #52            "R"
CB10  F0 0E      BEQ   CB20           ugrás a Memory Read-hoz
CB12  20 58 F2   JSR   F258           /RTS/
CB15  C9 57      CMP   #57            "W"
CB17  F0 37      BEQ   CB50           Memory Write-hoz
CB19  C9 45      CMP   #45            "E"
CB1B  D0 2E      BNE   CB4B
CB1D  6C 6F 00   JMP   (006F)         Memory Execute-hoz

******************************        M-R "Memory Read"
CB20  B1 6F      LDA   (6F),Y         byte olvasása
CB22  85 85      STA   85
CB24  AD 74 02   LDA   0274           utasítás-sor hossza
CB27  C9 06      CMP   #06            kisebb 6-nál
CB29  90 1A      BCC   CB45           igen
CB2B  AE 05 02   LDX   0205           szám /darabszám/
CB2E  CA         DEX
CB2F  F0 14      BEQ   CB45           csak egy byte?
CB31  8A         TXA                  byte-ok száma
CB32  18         CLC
CB33  65 6F      ADC   6F             plusz a kezdőcím
CB35  E6 6F      INC   6F
CB37  8D 49 02   STA   0249           végmutató
CB3A  A5 6F      LDA   6F
CB3C  85 A5      STA   A5             hibaüzenet puffer mutatója
CB3E  A5 70      LDA   70             a M-R kezdőcímére
CB40  85 A6      STA   A6
CB42  4C 43 D4   JMP   D443           a byte kiírása

CB45  20 EB D0   JSR   D0EB           olvasócsatorna megnyitása
CB48  4C 3A D4   JMP   D43A           byte-ok kiírása

CB4B  A9 31      LDA   #31
CB4D  4C C8 C1   JMP   C1C8           31, "syntax error"

******************************        M-W "memory-write"
CB50  B9 06 02   LDA   0206,Y         karakter /karakterek/ olvasása
CB53  91 6F      STA   (6F),Y         és tárolása
CB55  C8         INY
CB56  CC 05 02   CPY   0205           karakterek száma
CB59  90 F5      BCC   CB50           már az összes karakter?
CB5B  60         RTS

******************************        User utasítás
CB5C  AC 01 02   LDY   0201           második karakter
CB5F  C0 30      CPY   #30            "0"
CB61  D0 09      BNE   CB6C           nem
CB63  A9 EA      LDA   #EA
CB65  85 6B      STA   6B             mutató az USER-címek táblázatán
CB67  A9 FF      LDA   #FF            $FFEA
CB69  85 6C      STA   6C
CB6B  60         RTS

CB6C  20 72 CB   JSR   CB72
CB6F  4C 94 C1   JMP   C194           kész, hibaüzenet előkészítése
```

```
CB72  88        DEY
CB73  98        TYA
CB74  29 0F     AND  #0F        szám
CB76  0A        ASL   A         kétszerese
CB77  A8        TAY
CB78  B1 6B     LDA  (6B),Y     a tábla az adatban lévő mutatónál
CB7A  85 75     STA   75
CB7C  C8        INY             cím a $75/$76 után
CB7D  B1 6B     LDA  (6B),Y
CB7F  85 76     STA   76
CB81  6C 75 00  JMP  (0075)     a művelet végrehajtása

****************************    közvetlen elérésű csatorna megnyitása "#"
CB84  AD 8E 02  LDA   028E      utolsó egységszám
CB87  85 7F     STA   7F        egységszám
CB89  A5 83     LDA   83        csatorna-szám
CB8B  48        PHA
CB8C  20 3D C6  JSR   C63D      egység ellenőrzése, esetleg inicializálás
CB8F  68        PLA
CB90  85 83     STA   83
CB92  AE 74 02  LDX   0274      file-név hossza
CB95  CA        DEX
CB96  D0 0D     BNE   CBA5      egynél nagyobb?
CB98  A9 01     LDA  #01
CB9A  20 E2 D1  JSR   D1E2      csatorna és puffer lefoglalása
CB9D  4C F1 CB  JMP   CBF1      kapcsolók beállítása, kész

CBA0  A9 70     LDA  #70
CBA2  4C C8 C1  JMP   C1C8      70, "no channel"

CBA5  A0 01     LDY  #01
CBA7  20 7C CC  JSR   CC7C      a puffer-szám betöltése
CBAA  AE 85 02  LDX   0285      puffer-szám
CBAD  E0 05     CPX  #05        nagyobb, 5-tel egyenlő?
CBAF  B0 EF     BCS   CBA0      70, "no channel"
CBB1  A9 00     LDA  #00
CBB3  85 6F     STA   6F
CBB5  85 70     STA   70
CBB7  38        SEC
CBB8  26 6F     ROL   6F        puffer megkeresése a foglaltsági regiszterben
CBBA  26 70     ROL   70
CBBC  CA        DEX
CBBD  10 F9     BPL   CBB8
CBBF  A5 6F     LDA   6F
CBC1  2D 4F 02  AND   024F
CBC4  D0 DA     BNE   CBA0      puffer foglalt?
CBC6  A5 70     LDA   70
CBC8  2D 50 02  AND   0250
CBCB  D0 D3     BNE   CBA0      puffer foglalt?
CBCD  A5 6F     LDA   6F
CBCF  0D 4F 02  ORA   024F
CBD2  8D 4F 02  STA   024F
CBD5  A5 70     LDA   70        puffer lefoglalása
CBD7  0D 50 02  ORA   0250      csatorna megkeresése és lefoglalása
CBDA  8D 50 02  STA   0250
CBDD  A9 00     LDA  #00
CBDF  20 E2 D1  JSR   D1E2
CBE2  A6 82     LDX   82        csatorna-szám
CBE4  AD 85 02  LDA   0285      puffer-szám
CBE7  95 A7     STA   A7,X
CBE9  AA        TAX
CBEA  A5 7F     LDA   7F        egységszám
CBEC  95 00     STA   00,X
CBEE  9D 5B 02  STA   025B,X
CBF1  A6 83     LDX   83        másodlagos cím
CBF3  BD 2B 02  LDA   022B,X
CBF6  09 40     ORA  #40        a READ és a WRITE kapcsolók beállítása
```

```
CBF8   9D 2B 02     STA   022B,X
CBFB   A4 82        LDY   82          csatorna-szám
CBFD   A9 FF        LDA   #FF
CBFF   99 44 02     STA   0244,Y      végmutató
CC02   A9 89        LDA   #89
CC04   99 F2 00     STA   00F2,Y      a READ és a WRITE kapcsolók beállítása
CC07   B9 A7 00     LDA   00A7,Y      puffer-szám
CC0A   99 3E 02     STA   023E,Y
CC0D   0A           ASL   A           2-szer /2-szeres/
CC0E   AA           TAX
CC0F   A9 01        LDA   #01
CC11   95 99        STA   99,X        puffer mutató egyesen
CC13   A9 0E        LDA   #0E
CC15   99 EC 00     STA   00EC,Y      közvetlen elérésű kapcsoló
CC18   4C 94 C1     JMP   C194        kész

******************************        B-utasítás, "Block"
CC1B   A0 00        LDY   #00
CC1D   A2 00        LDX   #00
CC1F   A9 2D        LDA   #2D         "-"
CC21   20 68 C2     JSR   C268        minusz jel keresése
CC24   D0 0A        BNE   CC30        megvan?
CC26   A9 31        LDA   #31
CC28   4C C8 C1     JMP   C1C8        31, "syntax error"

CC2B   A9 30        LDA   #30
CC2D   4C C8 C1     JMP   C1C8        30, "syntax error"

CC30   8A           TXA
CC31   D0 F8        BNE   CC2B        ha vessző, akkor hiba!
CC33   A2 05        LDX   #05
CC35   B9 00 02     LDA   0200,Y      pufferből kapott karakter
CC38   DD 5D CC     CMP   CC5D,X      összehasonlítása az "AFRWP"-el
CC3B   F0 05        BEQ   CC42        megvan?
CC3D   CA           DEX
CC3E   10 F8        BPL   CC38        összehasonlitás az összes többi karakterrel,
CC40   30 E4        BMI   CC26        ha nincs meg, akkor hiba!
CC42   8A           TXA
CC43   09 80        ORA   #80         utasítás-szám,beállítása a 7-es bitnek
CC45   8D 2A 02     STA   022A
CC48   20 6F CC     JSR   CC6F        a paraméter betöltése
CC4B   AD 2A 02     LDA   022A
CC4E   0A           ASL   A           a szám 2-szer nagyobb
CC4F   AA           TAX               az indexnél
CC50   BD 64 CC     LDA   CC64,X      utasításcím /Hi/
CC53   85 70        STA   70
CC55   BD 63 CC     LDA   CC63,X      utasításcím /Lo/
CC58   85 6F        STA   6F
CC5A   6C 6F 00     JMP   (006F)      ugrás az utasításhoz

******************************        különböző "AFRWEP" blokk-utasítások nevei
CC5D   41 46 52 57 45 50

******************************        blokk-utasítások címei
CC63   03 CD                          $CD03, B-A
CC65   F5 CC                          $CCF5, B-F
CC67   56 CD                          $CD56, B-R
CC69   73 CD                          $CD73, B-W
CC6B   A3 CD                          $CDA3, B-E
CC6D   BD CD                          $CDBD, B-P

******************************        a blokk-utasítások paramétereinek betöltése
CC6F   A0 00        LDY   #00
CC71   A2 00        LDX   #00
CC73   A9 3A        LDA   #3A         ":"
CC75   20 68 C2     JSR   C268        a sorok kettőspontig tesztelése
CC78   D0 02        BNE   CC7C        megvan?
```

```
CC7A  A0 03     LDY  #03      nincs, a 4. karaktertől kezdeni
CC7C  B9 00 02  LDA  0200,Y   megkeresni az elválasztójelet!
CC7F  C9 20     CMP  #20      ' ' space
CC81  F0 08     BEQ  CC8B
CC83  C9 1D     CMP  #1D      kurzor jobbra
CC85  F0 04     BEQ  CC8B     ","
CC87  C9 2C     CMP  #2C
CC89  D0 07     BNE  CC92
CC8B  C8        INY
CC8C  CC 74 02  CPY  0274     sor vége?
CC8F  90 EB     BCC  CC7C
CC91  60        RTS

CC92  20 A1 CC  JSR  CCA1     a következő paraméter átvitele!
CC95  EE 77 02  INC  0277     a paraméter-számláló növelése
CC98  AC 79 02  LDY  0279
CC9B  E0 04     CPX  #04      összehasonlítás
CC9D  90 EC     BCC  CC8B     még nem lépte túl a maximális számot
CC9F  B0 8A     BCS  CC2B     3Ø, "syntax error"
CCA1  A9 00     LDA  #00
CCA3  85 6F     STA  6F
CCA5  85 70     STA  70
CCA7  85 72     STA  72
CCA9  A2 FF     LDX  #FF
CCAB  B9 00 02  LDA  0200,Y   a karakter betöltése az input-pufferből
CCAE  C9 40     CMP  #40
CCB0  B0 18     BCS  CCCA     nem számjegy?
CCB2  C9 30     CMP  #30      "Ø"
CCB4  90 14     BCC  CCCA     nem számjegy?
CCB6  29 0F     AND  #0F      konvertálás hexadecimálisra
CCB8  48        PHA
CCB9  A5 70     LDA  70
CCBB  85 71     STA  71       eltolás
CCBD  A5 6F     LDA  6F
CCBF  85 70     STA  70
CCC1  68        PLA
CCC2  85 6F     STA  6F       olvasott szám tárolása
CCC4  C8        INY           mutató növelése
CCC5  CC 74 02  CPY  0274     sor vége?
CCC8  90 E1     BCC  CCAB     nem
CCCA  8C 79 02  STY  0279     mutató tárolása
CCCD  18        CLC
CCCE  A9 00     LDA  #00
CCD0  E8        INX
CCD1  E0 03     CPX  #03
CCD3  B0 0F     BCS  CCE4     hexadecimális átalakitás
CCD5  B4 6F     LDY  6F,X
CCD7  88        DEY
CCD8  30 F6     BMI  CCD0
CCDA  7D F2 CC  ADC  CCF2,X   decimális érték hozzáadása
CCDD  90 F8     BCC  CCD7
CCDF  18        CLC
CCE0  E6 72     INC  72
CCE2  D0 F3     BNE  CCD7
CCE4  48        PHA
CCE5  AE 77 02  LDX  0277     paraméter-számláló
CCE8  A5 72     LDA  72
CCEA  9D 80 02  STA  0280,X   Hi-byte
CCED  68        PLA
CCEE  9D 85 02  STA  0285,X   Lo-byte
CCF1  60        RTS

******************************  1, 1Ø, 1ØØ decimális
CCF2  01 0A 64                  érték

******************************  B-F utasítás, "Block free"
CCF5  20 F5 CD  JSR  CDF5     a sáv, szektor és egységszám betöltése
```

```
CCF8  20 5F EF   JSR  EF5F      a blokk felszabadítása
CCFB  4C 94 C1   JMP  C194      kész, hibaüzenet előkészítése

*****************************
CCFE  A9 01      LDA  #01
CD00  8D F9 02   STA  02F9
*****************************   B-A utasítás, "Block Allocate"
CD03  20 F5 CD   JSR  CDF5      sáv, szektor és az egységszám betöltése
CD06  A5 81      LDA  81        szektor
CD08  48         PHA            tárolása
CD09  20 FA F1   JSR  F1FA      a blokk keresése a BAM-ban
CD0C  F0 0B      BEQ  CD19      foglalt a blokk?
CD0E  68         PLA            a kívánt szektor
CD0F  C5 81      CMP  81        azonos a következő szabad szektorral?
CD11  D0 19      BNE  CD2C      nem
CD13  20 90 EF   JSR  EF90      blokk lefoglalása a BAM-ban
CD16  4C 94 C1   JMP  C194      kész

CD19  68         PLA
CD1A  A9 00      LDA  #00
CD1C  85 81      STA  81        Ø szektor
CD1E  E6 80      INC  80        következő sáv
CD20  A5 80      LDA  80        sáv-szám
CD22  CD D7 FE   CMP  FED7      36, utolsó sáv-szám + 1
CD25  B0 0A      BCS  CD31      ha nagyobb vagy egyenlő "no block"
CD27  20 FA F1   JSR  F1FA      szabad blokk keresése a következő sávon
CD2A  F0 EE      BEQ  CD1A      ha nincs, a következő sáv ellenőrzése
CD2C  A9 65      LDA  #65
CD2E  20 45 E6   JSR  E645      65, "no block" következő szabad blokk
CD31  A9 65      LDA  #65
CD33  20 C8 C1   JSR  C1C8      65, "no block" nincs több szabad blokk

*****************************   csatorna megnyitása, paraméter
CD36  20 F2 CD   JSR  CDF2      beállítása
CD39  4C 60 D4   JMP  D460      blokk olvasása a lemezről

*****************************   egy byte betöltése a pufferből
CD3C  20 2F D1   JSR  D12F      mutató beállítása a pufferen belül
CD3F  A1 99      LDA  (99,X)    a byte betöltése
CD41  60         RTS

*****************************   egy blokk olvasása lemezről
CD42  20 36 CD   JSR  CD36      csatorna megnyitása, blokk olvasása
CD45  A9 00      LDA  #00
CD47  20 C8 D4   JSR  D4C8      puffer mutató nullára állítása
CD4A  20 3C CD   JSR  CD3C      egy byte beolvasása a pufferből
CD4D  99 44 02   STA  0244,Y
CD50  A9 89      LDA  #89       az irás- és olvasás-kapcsoló beállítása
CD52  99 F2 00   STA  00F2,Y
CD55  60         RTS

*****************************   B-R utasítás, "Block Read"
CD56  20 42 CD   JSR  CD42      egy blokk olvasása lemezről
CD59  20 EC D3   JSR  D3EC      a byte előkészítése
CD5C  4C 94 C1   JMP  C194      a hibaüzenet előkészítése

*****************************   U1 utasítás, "Block Read" helyettesítése
CD5F  20 6F CC   JSR  CC6F      utasítás paraméterének beolvasása
CD62  20 42 CD   JSR  CD42      blokk olvasása lemezről
CD65  B9 44 02   LDA  0244,Y    végmutató tárolása
CD68  99 3E 02   STA  023E,Y    adatbyte-ként
CD6B  A9 FF      LDA  #FF
CD6D  99 44 02   STA  0244,Y    végmutató $FF-en
CD70  4C 94 C1   JMP  C194      kész, hibaüzenet előkészítése
```

```
******************************          B-W utasítás, "Block Write"
CD73  20 F2 CD   JSR   CDF2    a csatorna megnyitása
CD76  20 E8 D4   JSR   D4E8    a puffer mutató beállítása
CD79  A8         TAY
CD7A  88         DEY
CD7B  C9 02      CMP   #02     puffer mutató 2-nél kisebb /Lo/
CD7D  B0 02      BCS   CD81    nem
CD7F  A0 01      LDY   #01
CD81  A9 00      LDA   #00
CD83  20 C8 D4   JSR   D4C8    puffer mutató nullán
CD86  98         TYA
CD87  20 F1 CF   JSR   CFF1    byte beírása a pufferbe
CD8A  8A         TXA
CD8B  48         PHA
CD8C  20 64 D4   JSR   D464    blokk felírása a lemezre
CD8F  68         PLA
CD90  AA         TAX
CD91  20 EE D3   JSR   D3EE    byte kihozása a pufferből
CD94  4C 94 C1   JMP   C194    kész, hibaüzenet

******************************          U2, "Block Write" helyett
CD97  20 6F CC   JSR   CC6F    utasítás paraméterének betöltése
CD9A  20 F2 CD   JSR   CDF2    csatorna megnyitása
CD9D  20 64 D4   JSR   D464    és a blokk beírása a lemezre
CDA0  4C 94 C1   JMP   C194    kész
                               "B-E" utasítás, "Block Execute"
CDA3  20 58 F2   JSR   F258    /RTS/
CDA6  20 36 CD   JSR   CD36    csatorna megnyitása és a blokk beolvasása
CDA9  A9 00      LDA   #00
CDAB  85 6F      STA   6F      cím /Lo/
CDAD  A6 F9      LDX   F9      puffer-szám
CDAF  BD E0 FE   LDA   FEE0,X  cím /Hi/
CDB2  85 70      STA   70
CDB4  20 BA CD   JSR   CDBA    ugrás a rutinhoz
CDB7  4C 94 C1   JMP   C194    kész

CDBA  6C 6F 00   JMP   (006F)  ugrás a rutinhoz

******************************          B-P utasítás, "Block Pointer"
CDBD  20 D2 CD   JSR   CDD2    csatorna megnyitása, puffer-szám betöltése
CDC0  A5 F9      LDA   F9      puffer-szám
CDC2  0A         ASL   A       * 2
CDC3  AA         TAX           indexként
CDC4  AD 86 02   LDA   0286    mutató-érték
CDC7  95 99      STA   99,X    puffer mutató tárolása
CDC9  20 2F D1   JSR   D12F    egy byte a pufferből
CDCC  20 EE D3   JSR   D3EE    egy byte előkészítése
CDCF  4C 94 C1   JMP   C194    kész

******************************          csatorna megnyitása
CDD2  A6 D3      LDX   D3
CDD4  E6 D3      INC   D3
CDD6  BD 85 02   LDA   0285,X  puffer-szám
CDD9  A8         TAY
CDDA  88         DEY
CDDB  88         DEY
CDDC  C0 0C      CPY   #0C     puffer-szám 14-nél kisebb
CDDE  90 05      BCC   CDE5    igen
CDE0  A9 70      LDA   #70
CDE2  4C C8 C1   JMP   C1C8    70, "no channel"

CDE5  85 83      STA   83      másodlagos cím
CDE7  20 EB D0   JSR   D0EB    csatorna megnyitása
CDEA  B0 F4      BCS   CDE0    ha már foglalt, 70 "no channel"
CDEC  20 93 DF   JSR   DF93    puffer-szám
CDEF  85 F9      STA   F9      beállítása
CDF1  60         RTS
```

```
*****************************
CDF2  20 D2 CD   JSR  CDD2      puffer-szám ellenőrzése, csatorna megnyitása
CDF5  A6 D3      LDX  D3        csatorna-szám
CDF7  BD 85 02   LDA  0285,X    puffer-cím
CDFA  29 01      AND  #01
CDFC  85 7F      STA  7F        egységszám
CDFE  BD 87 02   LDA  0287,X
CE01  85 81      STA  81        szektor
CE03  BD 86 02   LDA  0286,X
CE06  85 80      STA  80        sáv
CE08  20 5F D5   JSR  D55F      sáv és szektor ok?
CE0B  4C 00 C1   JMP  C100      LED bekapcsolása

*****************************   REL-file mutatójának beállitása
CE0E  20 2C CE   JSR  CE2C      rekord-szám * rekordhosszúság osztva
CE11  20 6E CE   JSR  CE6E      254-gyel = adatblokkok száma
CE14  A5 90      LDA  90        osztás maradéka = adatblokkban lévő mutatóval
CE16  85 D7      STA  D7        adat-mutató
CE18  20 71 CE   JSR  CE71
CE1B  E6 D7      INC  D7
CE1D  E6 D7      INC  D7        adatmutató plusz 2 /sáv, szektor-mutató!/
CE1F  A5 8B      LDA  8B        az osztás eredménye
CE21  85 D5      STA  D5        egyenlő az oldal-szektor számmal
CE23  A5 90      LDA  90        az osztás maradéka
CE25  0A         ASL  A         2-szer
CE26  18         CLC
CE27  69 10      ADC  #10       plusz 16
CE29  85 D6      STA  D6        egyenlő az adatblokkon az oldal-
CE2B  60         RTS            szektorban lévő mutatóval

*****************************
CE2C  20 D9 CE   JSR  CED9      operatív tár törlése
CE2F  85 92      STA  92
CE31  A6 82      LDX  82        csatorna-szám
CE33  B5 B5      LDA  B5,X      rekord-szám /Lo/
CE35  85 90      STA  90
CE37  B5 BB      LDA  BB,X      rekord-szám /Hi/
CE39  85 91      STA  91
CE3B  D0 04      BNE  CE41
CE3D  A5 90      LDA  90
CE3F  F0 0B      BEQ  CE4C      nem egyenlő nullával a rekord-szám?
CE41  A5 90      LDA  90
CE43  38         SEC
CE44  E9 01      SBC  #01       ha nem, egyet levonni
CE46  85 90      STA  90
CE48  B0 02      BCS  CE4C
CE4A  C6 91      DEC  91
CE4C  B5 C7      LDA  C7,X      rekordhosszúság
CE4E  85 6F      STA  6F
CE50  46 6F      LSR  6F
CE52  90 03      BCC  CE57
CE54  20 ED CE   JSR  CEED      rekord-szám * rekordhosszúság
CE57  20 E5 CE   JSR  CEE5      regiszter balra léptetése
CE5A  A5 6F      LDA  6F
CE5C  D0 F2      BNE  CE50
CE5E  A5 D4      LDA  D4
CE60  18         CLC
CE61  65 8B      ADC  8B
CE63  85 8B      STA  8B
CE65  90 06      BCC  CE6D      eredmény a $8B/$8C/$8D-ben
CE67  E6 8C      INC  8C
CE69  D0 02      BNE  CE6D
CE6B  E6 8D      INC  8D
CE6D  60         RTS

*****************************   osztás 254-gyel, adatblokk-szám kiszámitásához
CE6E  A9 FE      LDA  #FE       254
CE70  2C
```

```
****************************            az oldal-szektor kiszámítása
CE71  A9 78      LDA  #78               120
CE73  85 6F      STA  6F                osztó
CE75  A2 03      LDX  #03
CE77  B5 8F      LDA  8F,X
CE79  48         PHA
CE7A  B5 8A      LDA  8A,X
CE7C  95 8F      STA  8F,X
CE7E  68         PLA
CE7F  95 8A      STA  8A,X
CE81  CA         DEX
CE82  D0 F3      BNE  CE77
CE84  20 D9 CE   JSR  CED9              az operativ tár törlése
CE87  A2 00      LDX  #00
CE89  B5 90      LDA  90,X
CE8B  95 8F      STA  8F,X
CE8D  E8         INX
CE8E  E0 04      CPX  #04
CE90  90 F7      BCC  CE89
CE92  A9 00      LDA  #00
CE94  85 92      STA  92
CE96  24 6F      BIT  6F
CE98  30 09      BMI  CEA3
CE9A  06 8F      ASL  8F
CE9C  08         PHP
CE9D  46 8F      LSR  8F
CE9F  28         PLP
CEA0  20 E6 CE   JSR  CEE6              1-es regiszter balra léptetése
CEA3  20 ED CE   JSR  CEED              0 regiszter tartalmának hozzáadása az 1-es-hez
CEA6  20 E5 CE   JSR  CEE5              1-es regiszter balra létetése
CEA9  24 6F      BIT  6F
CEAB  30 03      BMI  CEB0
CEAD  20 E2 CE   JSR  CEE2              1-es regiszter kétszeri balra léptetése
CEB0  A5 8F      LDA  8F
CEB2  18         CLC
CEB3  65 90      ADC  90
CEB5  85 90      STA  90
CEB7  90 06      BCC  CEBF
CEB9  E6 91      INC  91
CEBB  D0 02      BNE  CEBF
CEBD  E6 92      INC  92
CEBF  A5 92      LDA  92
CEC1  05 91      ORA  91
CEC3  D0 C2      BNE  CE87
CEC5  A5 90      LDA  90
CEC7  38         SEC
CEC8  E5 6F      SBC  6F                hányados a $8B/$8C/$8D-ben
CECA  90 0C      BCC  CED8
CECC  E6 8B      INC  8B
CECE  D0 06      BNE  CED6
CED0  E6 8C      INC  8C
CED2  D0 02      BNE  CED6
CED4  E6 8D      INC  8D
CED6  85 90      STA  90                maradék a $90-ben
CED8  60         RTS

****************************            operativ tár törlése
CED9  A9 00      LDA  #00
CEDB  85 8B      STA  8B
CEDD  85 8C      STA  8C
CEDF  85 8D      STA  8D
CEE1  60         RTS

****************************            a regiszter kétszeri balra léptetése
CEE2  20 E5 CE   JSR  CEE5
```

```
******************************      a regiszter egyszeri balra léptetése
CEE5  18          CLC
CEE6  26 90       ROL   90
CEE8  26 91       ROL   91
CEEA  26 92       ROL   92
CEEC  60          RTS

CEED  18          CLC
CEEE  A2 FD       LDX   #FD
CEF0  B5 8E       LDA   8E,X        $90/$91/$92 regiszter tartalmána hozzá-
CEF2  75 93       ADC   93,X        adása a $8B/$8C/$8D regiszterhez
CEF4  95 8E       STA   8E,X
CEF6  E8          INX
CEF7  D0 F7       BNE   CEF0
CEF9  60          RTS

CEFA  A2 00       LDX   #00
CEFC  8A          TXA
CEFD  95 FA       STA   FA,X
CEFF  E8          INX
CF00  E0 04       CPX   #04
CF02  D0 F8       BNE   CEFC
CF04  A9 06       LDA   #06
CF06  95 FA       STA   FA,X
CF08  60          RTS

CF09  A0 04       LDY   #04
CF0B  A6 82       LDX   82          csatorna-szám
CF0D  B9 FA 00    LDA   00FA,Y
CF10  96 FA       STX   FA,Y
CF12  C5 82       CMP   82          csatorna-szám
CF14  F0 07       BEQ   CF1D
CF16  88          DEY
CF17  30 E1       BMI   CEFA
CF19  AA          TAX
CF1A  4C 0D CF    JMP   CF0D

CF1D  60          RTS

CF1E  20 09 CF    JSR   CF09
CF21  20 B7 DF    JSR   DFB7
CF24  D0 46       BNE   CF6C
CF26  20 D3 D1    JSR   D1D3        egységszám beállítása
CF29  20 8E D2    JSR   D28E
CF2C  30 48       BMI   CF76
CF2E  20 C2 DF    JSR   DFC2
CF31  A5 80       LDA   80          sáv
CF33  48          PHA
CF34  A5 81       LDA   81          szektor
CF36  48          PHA
CF37  A9 01       LDA   #01
CF39  20 F6 D4    JSR   D4F6        1-es byte a pufferból
CF3C  85 81       STA   81          szektor
CF3E  A9 00       LDA   #00
CF40  20 F6 D4    JSR   D4F6        0. byte a pufferból
CF43  85 80       STA   80          sáv
CF45  F0 1F       BEQ   CF66
CF47  20 25 D1    JSR   D125        file-tipus ellenőrzése
CF4A  F0 0B       BEQ   CF57        relatív file?
CF4C  20 AB DD    JSR   DDAB
CF4F  D0 06       BNE   CF57
CF51  20 8C CF    JSR   CF8C
CF54  4C 5D CF    JMP   CF5D

CF57  20 8C CF    JSR   CF8C
CF5A  20 57 DE    JSR   DE57
CF5D  68          PLA
```

```
CF5E  85 81     STA  81      szektor-
CF60  68        PLA
CF61  85 80     STA  80      és a sáv-szám visszatöltése
CF63  4C 6F CF  JMP  CF6F

CF66  68        PLA
CF67  85 81     STA  81      szektor-
CF69  68        PLA
CF6A  85 80     STA  80      és a sáv-szám visszatöltése
CF6C  20 8C CF  JSR  CF8C
CF6F  20 93 DF  JSR  DF93
CF72  AA        TAX
CF73  4C 99 D5  JMP  D599    és ellenőrzése

CF76  A9 70     LDA  #70
CF78  4C C8 C1  JMP  C1C8    70, "no channel"

CF7B  20 09 CF  JSR  CF09
CF7E  20 B7 DF  JSR  DFB7
CF81  D0 08     BNE  CF8B
CF83  20 8E D2  JSR  D28E
CF86  30 EE     BMI  CF76
CF88  20 C2 DF  JSR  DFC2
CF8B  60        RTS

*****************************  puffer váltás
CF8C  A6 82     LDX  82      csatorna-szám
CF8E  B5 A7     LDA  A7,X
CF90  49 80     EOR  #80
CF92  95 A7     STA  A7,X
CF94  B5 AE     LDA  AE,X    a 7-es bit megfordítása a táblázatban
CF96  49 80     EOR  #80
CF98  95 AE     STA  AE,X
CF9A  60        RTS

*****************************  adatbyte beírása a pufferbe
CF9B  A2 12     LDX  #12     18-as csatorna
CF9D  86 83     STX  83
CF9F  20 07 D1  JSR  D107    írás-csatorna megnyitása
CFA2  20 00 C1  JSR  C100    LED bekapcsolása
CFA5  20 25 D1  JSR  D125    file-tipus ellenőrzése
CFA8  90 05     BCC  CFAF    nem relatív file
CFAA  A9 20     LDA  #20
CFAC  20 9D DD  JSR  DD9D    puffer váltás
CFAF  A5 83     LDA  83      másodlagos cím
CFB1  C9 0F     CMP  #0F     15 ?
CFB3  F0 23     BEQ  CFD8    igen
CFB5  D0 08     BNE  CFBF    nem
CFB7  A5 84     LDA  84      másodlagos cím
CFB9  29 8F     AND  #8F
CFBB  C9 0F     CMP  #0F     nagyobb vagy egyenlő mint 15?
CFBD  B0 19     BCS  CFD8    ha igen, - input-puffer
CFBF  20 25 D1  JSR  D125    file-tipus ellenőrzése
CFC2  B0 05     BCS  CFC9    relatív file vagy közvetlen elérés?
CFC4  A5 85     LDA  85      adatbyte
CFC6  4C 9D D1  JMP  D19D    betöltése a pufferbe

CFC9  D0 03     BNE  CFCE    közvetlen elérésű file?
CFCB  4C AB E0  JMP  E0AB    adatbyte-ok beírása a relatív file-ba

CFCE  A5 85     LDA  85
CFD0  20 F1 CF  JSR  CFF1    adatbyte beírása a pufferbe
CFD3  A4 82     LDY  82      következő byte-csatorna számának
CFD5  4C EE D3  JMP  D3EE    előkészítése
```

```
CFD8  A9 04     LDA  #04     4-es csatorna
CFDA  85 82     STA  82      megfelel az input-puffernek
CFDC  20 E8 D4  JSR  D4E8    a puffer mutató beállítása
CFDF  C9 2A     CMP  #2A     4Ø
CFE1  F0 05     BEQ  CFE8    puffer vége?
CFE3  A5 85     LDA  85
CFE5  20 F1 CF  JSR  CFF1    adatbyte beírása a pufferbe
CFE8  A5 F8     LDA  F8      be van állítva a kapcsoló?
CFEA  F0 01     BEQ  CFED    igen
CFEC  60        RTS

CFED  EE 55 02  INC  0255    az utasítás-kapcsoló beállítása
CFF0  60        RTS

*****************************  adatbyte beírása a pufferbe
CFF1  48        PHA          adatbyte tárolása
CFF2  20 93 DF  JSR  DF93    puffer-szám betöltése
CFF5  10 06     BPL  CFFD    hozzá van rendelve a puffer?
CFF7  68        PLA
CFF8  A9 61     LDA  #61
CFFA  4C C8 C1  JMP  C1C8    61, "file not open"

CFFD  0A        ASL  A       puffer-szám kétszer nagyobb
CFFE  AA        TAX          az indexnél
CFFF  68        PLA          adatbyte
D000  81 99     STA  (99,X)  beírása a pufferbe
D002  F6 99     INC  99,X    puffer mutató növelése
D004  60        RTS

*****************************  I-utasítás, inicializálás
D005  20 D1 C1  JSR  C1D1    egységszám megkeresése
D008  20 42 D0  JSR  D042    BAM betöltése
D00B  4C 94 C1  JMP  C194    lemez-státus előkészítése

*****************************
D00E  20 0F F1  JSR  F10F
D011  A8        TAY
D012  B6 A7     LDX  A7,Y
D014  E0 FF     CPX  #FF
D016  D0 14     BNE  D02C
D018  48        PHA
D019  20 8E D2  JSR  D28E
D01C  AA        TAX
D01D  10 05     BPL  D024
D01F  A9 70     LDA  #70
D021  20 48 E6  JSR  E648    70, "no channel"
D024  68        PLA
D025  A8        TAY
D026  8A        TXA
D027  09 80     ORA  #80
D029  99 A7 00  STA  00A7,Y
D02C  8A        TXA
D02D  29 0F     AND  #0F
D02F  85 F9     STA  F9
D031  A2 00     LDX  #00
D033  86 81     STX  81      Ø szektor
D035  AE 85 FE  LDX  FE85    18
D038  86 80     STX  80      18-as sáv
D03A  20 D3 D6  JSR  D6D3    paraméter átadása a lemezvezérlőnek
D03D  A9 B0     LDA  #B0     "Block Header lesen" utasításkód átadása
D03F  4C 8C D5  JMP  D58C    a lemezvezérlőnek

*****************************  BAM betöltése
D042  20 D1 F0  JSR  F0D1
D045  20 13 D3  JSR  D313
D048  20 0E D0  JSR  D00E    blokk olvasása
D04B  A6 7F     LDX  7F      egységszám
```

```
D04D  A9 00      LDA  #00
D04F  9D 51 02   STA  0251,X     a "BAM megváltozott" kapcsoló visszaállítása
D052  8A         TXA
D053  0A         ASL  A
D054  AA         TAX
D055  A5 16      LDA  16
D057  95 12      STA  12,X
D059  A5 17      LDA  17         ID tárolása
D05B  95 13      STA  13,X
D05D  20 86 D5   JSR  D586
D060  A5 F9      LDA  F9         puffer-szám
D062  0A         ASL  A
D063  AA         TAX
D064  A9 02      LDA  #02        puffer mutató $200-on
D066  95 99      STA  99,X
D068  A1 99      LDA  (99,X)     egy karakter betöltése a pufferből
D06A  A6 7F      LDX  7F         egységszám
D06C  9D 01 01   STA  0101,X
D06F  A9 00      LDA  #00
D071  95 1C      STA  1C,X       irásvédelem-kapcsoló
D073  95 FF      STA  FF,X       olvasáshiba-kapcsoló

******************************   szabad blokkok kiszámítása
D075  20 3A EF   JSR  EF3A       a $6D/$6E utáni puffer-cím
D078  A0 04      LDY  #04        a 4-es pozíciótól kezdve
D07A  A9 00      LDA  #00
D07C  AA         TAX
D07D  18         CLC
D07E  71 6D      ADC  (6D),Y     sávonkénti üres bblokkok számának hozzáadása
D080  90 01      BCC  D083
D082  E8         INX             X, mint Hi byte
D083  C8         INY
D084  C8         INY             plusz 4
D085  C8         INY
D086  C8         INY
D087  C0 48      CPY  #48        18-as sáv?
D089  F0 F8      BEQ  D083       ha igen, kihagyni!
D08B  C0 90      CPY  #90        utolsó sáv-szám?
D08D  D0 EE      BNE  D07D       nem
D08F  48         PHA             Lo byte
D090  8A         TXA             Hi byte
D091  A6 7F      LDX  7F         egységszám
D093  9D FC 02   STA  02FC,X     $2FC utáni Hi byte
D096  68         PLA             Lo byte
D097  9D FA 02   STA  02FA,X     $2FA után
D09A  60         RTS

******************************
D09B  20 D0 D6   JSR  D6D0       paraméter a lemezvezérlőn
D09E  20 C3 D0   JSR  D0C3       blokk olvasása
D0A1  20 99 D5   JSR  D599       ok?
D0A4  20 37 D1   JSR  D137       a byte betöltése a pufferből
D0A7  85 80      STA  80         következő byte sávja
D0A9  20 37 D1   JSR  D137       a pufferből
D0AC  85 81      STA  81         szektor
D0AE  60         RTS

D0AF  20 9B D0   JSR  D09B
D0B2  A5 80      LDA  80         sáv
D0B4  D0 01      BNE  D0B7
D0B6  60         RTS

D0B7  20 1E CF   JSR  CF1E       puffer váltás
D0BA  20 D0 D6   JSR  D6D0       paraméter a lemezvezérlőn
D0BD  20 C3 D0   JSR  D0C3       blokk olvasása
D0C0  4C 1E CF   JMP  CF1E       puffer váltás
```

```
******************************         blokk olvasása
D0C3  A9 80      LDA  #80              olvasás-kód
D0C5  D0 02      BNE  D0C9

******************************         blokk beírása
D0C7  A9 90      LDA  #90              írás-kód
D0C9  8D 4D 02   STA  024D             tárolása
D0CC  20 93 DF   JSR  DF93             a puffer-szám beolvasása
D0CF  AA         TAX
D0D0  20 06 D5   JSR  D506             sáv/szektor betöltése, blokk olvasása,írása
D0D3  8A         TXA
D0D4  48         PHA
D0D5  0A         ASL  A                puffer mutató kétszer
D0D6  AA         TAX
D0D7  A9 00      LDA  #00
D0D9  95 99      STA  99,X             mutató a pufferben nullán
D0DB  20 25 D1   JSR  D125             a file-tipus beolvasása
D0DE  C9 04      CMP  #04              relativ file, vagy közvetlen elérés?
D0E0  B0 06      BCS  D0E8             igen
D0E2  F6 B5      INC  B5,X
D0E4  D0 02      BNE  D0E8             a blokk-számláló növelése
D0E6  F6 BB      INC  BB,X
D0E8  68         PLA
D0E9  AA         TAX
D0EA  60         RTS

******************************         olvasás-csatorna megnyitása
D0EB  A5 83      LDA  83               másodlagos cím
D0ED  C9 13      CMP  #13              19
D0EF  90 02      BCC  D0F3             kisebb?
D0F1  29 0F      AND  #0F
D0F3  C9 0F      CMP  #0F
D0F5  D0 02      BNE  D0F9
D0F7  A9 10      LDA  #10              16
D0F9  AA         TAX
D0FA  38         SEC
D0FB  BD 2B 02   LDA  022B,X
D0FE  30 06      BMI  D106
D100  29 0F      AND  #0F
D102  85 82      STA  82
D104  AA         TAX
D105  18         CLC                   ok kapcsoló
D106  60         RTS

******************************         írás-csatorna megnyitása
D107  A5 83      LDA  83               máodlagos cím
D109  C9 13      CMP  #13              19
D10B  90 02      BCC  D10F             kisebb?
D10D  29 0F      AND  #0F
D10F  AA         TAX
D110  BD 2B 02   LDA  022B,X           csatorna-szám
D113  A8         TAY
D114  0A         ASL  A
D115  90 0A      BCC  D121
D117  30 0A      BMI  D123
D119  98         TYA
D11A  29 0F      AND  #0F
D11C  85 82      STA  82
D11E  AA         TAX
D11F  18         CLC                   ok kapcsoló
D120  60         RTS

D121  30 F6      BMI  D119
D123  38         SEC                   csatorna-kapcsoló foglalt
D124  60         RTS
```

```
******************************     REL-file-tipus ellenőrzése
D125  A6 B2     LDX  B2
D127  B5 EC     LDA  EC,X
D129  4A        LSR  A
D12A  29 07     AND  #07
D12C  C9 04     CMP  #04          "REL"?
D12E  60        RTS

******************************     puffer- és csatorna-szám beolvasása
D12F  20 93 DF  JSR  DF93         puffer-szám beolvasása
D132  0A        ASL  A
D133  AA        TAX
D134  A4 B2     LDY  B2
D136  60        RTS

******************************     egy byte beolvasása a pufferból
D137  20 2F D1  JSR  D12F         puffer- és csatorna-szám beolvasása
D13A  B9 44 02  LDA  0244,Y       végmutató
D13D  F0 12     BEQ  D151
D13F  A1 99     LDA  (99,X)       byte beolvasása a pufferból
D141  48        PHA
D142  B5 99     LDA  99,X         puffer mutató
D144  D9 44 02  CMP  0244,Y       azonos a végmutatóval?
D147  D0 04     BNE  D14D         nem
D149  A9 FF     LDA  #FF
D14B  95 99     STA  99,X         puffer mutató /1/
D14D  68        PLA               adatbyte
D14E  F6 99     INC  99,X         puffer mutató növelése
D150  60        RTS

D151  A1 99     LDA  (99,X)       egy karakter betöltése a pufferból
D153  F6 99     INC  99,X         a puffer mutató növelése
D155  60        RTS

******************************     egy byte beolvasása, a következő blokk
D156  20 37 D1  JSR  D137         byte beolvasása a pufferból
D159  D0 36     BNE  D191         nem az utolsó karakter?
D15B  85 85     STA  85           adatbyte tárolása
D15D  B9 44 02  LDA  0244,Y       végmutató
D160  F0 08     BEQ  D16A         igen
D162  A9 80     LDA  #80
D164  99 F2 00  STA  00F2,Y       READ-kapcsoló
D167  A5 85     LDA  85           adatbyte
D169  60        RTS

D16A  20 1E CF  JSR  CF1E         puffer váltás, következő blokk olvasása
D16D  A9 00     LDA  #00
D16F  20 C8 D4  JSR  D4C8         puffer mutató nullára állítása
D172  20 37 D1  JSR  D137         első byte a pufferból
D175  C9 00     CMP  #00          sáv-szám nulla?
D177  F0 19     BEQ  D192         ha igen, az utolsó blokk
D179  85 80     STA  80           sáv-szám tárolása
D17B  20 37 D1  JSR  D137         utolsó byte, olvasása, tárolása
D17E  85 81     STA  81           láncolt szektoronként
D180  20 1E CF  JSR  CF1E         puffer váltás, következő blokk olvasása
D183  20 D3 D1  JSR  D1D3         egységszám tárolása
D186  20 D0 D6  JSR  D6D0         paraméter a lemezvezérlőn
D189  20 C3 D0  JSR  D0C3         az olvasási utasítás átadása
D18C  20 1E CF  JSR  CF1E         puffer váltás, a következő blokk olvasása
D18F  A5 85     LDA  85           adatbyte visszatöltése
D191  60        RTS

D192  20 37 D1  JSR  D137         következő byte beolvasása
D195  A4 B2     LDY  B2
D197  99 44 02  STA  0244,Y       tárolása végmutatóként
D19A  A5 85     LDA  85           adatbyte visszatöltése
D19C  60        RTS
```

```
******************************       pufferban lévő byte és blokk felírása
D19D   20 F1 CF     JSR    CFF1      byte a pufferban
D1A0   F0 01        BEQ    D1A3      megtelt a puffer?
D1A2   60           RTS

D1A3   20 D3 D1     JSR    D1D3      egységszám beolvasása
D1A6   20 1E F1     JSR    F11E      szabad blokk keresése a BAM-ban
D1A9   A9 00        LDA    #00
D1AB   20 C8 D4     JSR    D4C8      puffer mutató nullán
D1AE   A5 80        LDA    80
D1B0   20 F1 CF     JSR    CFF1      sáv-szám, mint első byte
D1B3   A5 81        LDA    81
D1B5   20 F1 CF     JSR    CFF1      szektor-szám, mint második byte
D1B8   20 C7 D0     JSR    D0C7      blokk írása
D1BB   20 1E CF     JSR    CF1E      puffer váltás
D1BE   20 D0 D6     JSR    D6D0      paraméter a lemezvezérlőn
D1C1   A9 02        LDA    #02
D1C3   4C C8 D4     JMP    D4C8      puffer mutató 2-n

******************************       a puffer mutató növelése
D1C6   85 6F        STA    6F
D1C8   20 E8 D4     JSR    D4E8      a puffer mutató beolvasása
D1CB   18           CLC
D1CC   65 6F        ADC    6F
D1CE   95 99        STA    99,X      és növelése
D1D0   85 94        STA    94
D1D2   60           RTS

******************************       egységszám beolvasása
D1D3   20 93 DF     JSR    DF93      puffer-szám beolvasása
D1D6   AA           TAX
D1D7   BD 5B 02     LDA    025B,X
D1DA   29 01        AND    #01       egységszám leválasztása
D1DC   85 7F        STA    7F        és tárolása
D1DE   60           RTS

******************************       az írás-csatorna és a puffer keresése
D1DF   38           SEC              írás-kapcsoló
D1E0   B0 01        BCS    D1E3

******************************       az olvasó-csatorna és a puffer keresése
D1E2   18           CLC              olvasás-kapcsoló
D1E3   08           PHP              tárolása
D1E4   85 6F        STA    6F        pufferek szám
D1E6   20 27 D2     JSR    D227      csatorna lezárása
D1E9   20 7F D3     JSR    D37F      szabad csatorna lefoglalása
D1EC   85 82        STA    82        csatorna-szám
D1EE   A6 83        LDX    83        másodlagos cím
D1F0   28           PLP
D1F1   90 02        BCC    D1F5      olvasás-csatorna?
D1F3   09 80        ORA    #80       olvasás-kapcsoló
D1F5   9D 2B 02     STA    022B,X    beállítása
D1F8   29 3F        AND    #3F
D1FA   A8           TAY
D1FB   A9 FF        LDA    #FF       alapértelmezés
D1FD   99 A7 00     STA    00A7,Y
D200   99 AE 00     STA    00AE,Y    beírása a hozzárendelési táblázatokba
D203   99 CD 00     STA    00CD,Y
D206   C6 6F        DEC    6F        a pufferek számának csökkentése
D208   30 1C        BMI    D226      már kész?
D20A   20 8E D2     JSR    D28E      puffer keresése
D20D   10 08        BPL    D217      van?
D20F   20 5A D2     JSR    D25A      kapcsolók törlése a táblázatban
D212   A9 70        LDA    #70
D214   4C C8 C1     JMP    C1C8      70, "no channel"
```

```
D217  99 A7 00    STA   00A7,Y       puffer-szám a táblázatban
D21A  C6 6F       DEC   6F           pufferek száma
D21C  30 08       BMI   D226         már kész?
D21E  20 8E D2    JSR   D28E         még nem sikerült
D221  30 EC       BMI   D20F         megtalálni?
D223  99 AE 00    STA   00AE,Y       puffer-szám a táblázatban
D226  60          RTS

****************************         csatorna lezárása
D227  A5 83       LDA   83           másodlagos cím
D229  C9 0F       CMP   #0F          15 ?
D22B  D0 01       BNE   D22E         nem
D22D  60          RTS                egyébként kész

D22E  A6 83       LDX   83
D230  BD 2B 02    LDA   022B,X       csatorna-szám
D233  C9 FF       CMP   #FF          nincs hozzárendelve?
D235  F0 22       BEQ   D259         ha nincs, kész
D237  29 3F       AND   #3F
D239  85 82       STA   82           csatorna-szám
D23B  A9 FF       LDA   #FF
D23D  9D 2B 02    STA   022B,X       hozzárendelés törlése a táblázatban
D240  A6 82       LDX   82
D242  A9 00       LDA   #00
D244  95 F2       STA   F2,X         a READ- és a WRITE-kapcsoló törlése
D246  20 5A D2    JSR   D25A         puffer felszabadítása
D249  A6 82       LDX   82           csatorna-szám
D24B  A9 01       LDA   #01          Ø bit beállítása
D24D  CA          DEX
D24E  30 03       BMI   D253         léptetés a helyes pozicióra
D250  0A          ASL   A
D251  D0 FA       BNE   D24D
D253  0D 56 02    ORA   0256         és a foglaltsági regiszter felszabadítása
D256  8D 56 02    STA   0256
D259  60          RTS

****************************         a puffer felszabadítása
D25A  A6 82       LDX   82           csatorna-szám
D25C  B5 A7       LDA   A7,X         puffer-szám
D25E  C9 FF       CMP   #FF
D260  F0 09       BEQ   D26B         nincs hozzárendelve?
D262  48          PHA
D263  A9 FF       LDA   #FF
D265  95 A7       STA   A7,X         a puffer-hozzárendelés törlése
D267  68          PLA
D268  20 F3 D2    JSR   D2F3         a puffer törlése a foglaltsági regiszterben
D26B  A6 82       LDX   82           csatorna-szám
D26D  B5 AE       LDA   AE,X
D26F  C9 FF       CMP   #FF          a második táblázatban van hozzárendelve?
D271  F0 09       BEQ   D27C         nem
D273  48          PHA
D274  A9 FF       LDA   #FF
D276  95 AE       STA   AE,X         hozzárendelés törlése
D278  68          PLA
D279  20 F3 D2    JSR   D2F3         puffer törlése a foglaltsági regiszterben
D27C  A6 82       LDX   82           csatorna-szám
D27E  B5 CD       LDA   CD,X
D280  C9 FF       CMP   #FF          a harmadik táblázatban van hozzárendelve?
D282  F0 09       BEQ   D28D         nem
D284  48          PHA
D285  A9 FF       LDA   #FF
D287  95 CD       STA   CD,X         hozzárendelés törlése
D289  68          PLA
D28A  20 F3 D2    JSR   D2F3         puffer törlése a foglaltsági regiszterben
D28D  60          RTS
```

```
******************************     puffer keresése
D28E  98        TYA
D28F  48        PHA
D290  A0 01     LDY  #01
D292  20 BA D2  JSR  D2BA
D295  10 0C     BPL  D2A3
D297  88        DEY
D298  20 BA D2  JSR  D2BA
D29B  10 06     BPL  D2A3
D29D  20 39 D3  JSR  D339
D2A0  AA        TAX
D2A1  30 13     BMI  D2B6
D2A3  B5 00     LDA  00,X
D2A5  30 FC     BMI  D2A3
D2A7  A5 7F     LDA  7F
D2A9  95 00     STA  00,X
D2AB  9D 5B 02  STA  025B,X
D2AE  8A        TXA
D2AF  0A        ASL  A
D2B0  A8        TAY
D2B1  A9 02     LDA  #02
D2B3  99 99 00  STA  0099,Y
D2B6  68        PLA
D2B7  A8        TAY
D2B8  8A        TXA
D2B9  60        RTS

D2BA  A2 07     LDX  #07
D2BC  B9 4F 02  LDA  024F,Y
D2BF  3D E9 EF  AND  EFE9,X     bit törlése
D2C2  F0 04     BEQ  D2C8
D2C4  CA        DEX
D2C5  10 F5     BPL  D2BC
D2C7  60        RTS

D2C8  B9 4F 02  LDA  024F,Y
D2CB  5D E9 EF  EOR  EFE9,X     bit megfordítása
D2CE  99 4F 02  STA  024F,Y
D2D1  8A        TXA             puffer-szám
D2D2  88        DEY
D2D3  30 03     BMI  D2D8
D2D5  18        CLC
D2D6  69 08     ADC  #08
D2D8  AA        TAX             puffer-szám
D2D9  60        RTS

D2DA  A6 82     LDX  82
D2DC  B5 A7     LDA  A7,X
D2DE  30 09     BMI  D2E9
D2E0  8A        TXA
D2E1  18        CLC
D2E2  69 07     ADC  #07
D2E4  AA        TAX
D2E5  B5 A7     LDA  A7,X
D2E7  10 F0     BPL  D2D9
D2E9  C9 FF     CMP  #FF
D2EB  F0 EC     BEQ  D2D9
D2ED  48        PHA
D2EE  A9 FF     LDA  #FF
D2F0  95 A7     STA  A7,X
D2F2  68        PLA
D2F3  29 0F     AND  #0F
D2F5  A8        TAY             puffer-szám
D2F6  C8        INY
D2F7  A2 10     LDX  #10        16
D2F9  6E 50 02  ROR  0250
```

```
D2FC  6E 4F 02   ROR  024F      16-bit, foglaltsági regiszter állásának táro-
D2FF  88         DEY            lása
D300  D0 01      BNE  D303
D302  18         CLC            puffer-bit törlése
D303  CA         DEX
D304  10 F3      BPL  D2F9
D306  60         RTS

****************************    összes csatorna lezárása
D307  A9 0E      LDA  #0E       14
D309  85 83      STA  83        másodlagos cím
D30B  20 27 D2   JSR  D227      csatorna lezárása
D30E  C6 83      DEC  83        következő másodlagos cím
D310  D0 F9      BNE  D30B
D312  60         RTS

****************************    a többi egység összes csatornájának lezárása
D313  A9 0E      LDA  #0E       14
D315  85 83      STA  83        másodlagos cím
D317  A6 83      LDX  83
D319  BD 2B 02   LDA  022B,X    hozzárendelési táblázat
D31C  C9 FF      CMP  #FF       hozzá van rendelve a csatorna?
D31E  F0 14      BEQ  D334      nincs
D320  29 3F      AND  #3F
D322  85 82      STA  82        csatorna-szám
D324  20 93 DF   JSR  DF93      puffer-szám betöltése
D327  AA         TAX
D328  BD 5B 02   LDA  025B,X    egységszám
D32B  29 01      AND  #01       leválasztása
D32D  C5 7F      CMP  7F        azonos az aktuális egységszámmal?
D32F  D0 03      BNE  D334      nem
D331  20 27 D2   JSR  D227      csatorna lezárása
D334  C6 83      DEC  83        következő csatorna
D336  10 DF      BPL  D317
D338  60         RTS

****************************
D339  A5 6F      LDA  6F
D33B  48         PHA
D33C  A0 00      LDY  #00
D33E  B6 FA      LDX  FA,Y
D340  B5 A7      LDA  A7,X
D342  10 04      BPL  D348
D344  C9 FF      CMP  #FF
D346  D0 16      BNE  D35E
D348  8A         TXA
D349  18         CLC
D34A  69 07      ADC  #07
D34C  AA         TAX
D34D  B5 A7      LDA  A7,X
D34F  10 04      BPL  D355
D351  C9 FF      CMP  #FF
D353  D0 09      BNE  D35E
D355  C8         INY
D356  C0 05      CPY  #05
D358  90 E4      BCC  D33E
D35A  A2 FF      LDX  #FF
D35C  D0 1C      BNE  D37A
D35E  86 6F      STX  6F
D360  29 3F      AND  #3F
D362  AA         TAX
D363  B5 00      LDA  00,X
D365  30 FC      BMI  D363
D367  C9 02      CMP  #02
D369  90 08      BCC  D373
D36B  A6 6F      LDX  6F
```

```
D36D  E0 07     CPX  #07
D36F  90 D7     BCC  D348
D371  B0 E2     BCS  D355
D373  A4 6F     LDY  6F
D375  A9 FF     LDA  #FF
D377  99 A7 00  STA  00A7,Y
D37A  68        PLA
D37B  85 6F     STA  6F
D37D  8A        TXA
D37E  60        RTS

****************************  csatorna keresése és lefoglalása
D37F  A0 00     LDY  #00
D381  A9 01     LDA  #01      a Ø. bit beállítása
D383  2C 56 02  BIT  0256
D386  D0 09     BNE  D391     szabad a csatorna?
D388  C8        INY
D389  0A        ASL  A        bit balra léptetése
D38A  D0 F7     BNE  D383     megtörtént az összes csatorna ellenőrzése?
D38C  A9 70     LDA  #70
D38E  4C C8 C1  JMP  C1C8     7Ø, "no channel"

D391  49 FF     EOR  #FF      bitminta megfordítása
D393  2D 56 02  AND  0256     bit törlése
D396  8D 56 02  STA  0256     csatorna lefoglalása
D399  98        TYA
D39A  60        RTS

****************************  a byte betöltése kiíráshoz
D39B  20 EB D0  JSR  D0EB     olvasás-csatorna megnyitása
D39E  20 00 C1  JSR  C100     LED bekapcsolása
D3A1  20 AA D3  JSR  D3AA     byte betöltése az output-regiszterbe
D3A4  A6 82     LDX  82       csatorna-szám
D3A6  BD 3E 02  LDA  023E,X   egy byte betöltése
D3A9  60        RTS

D3AA  A6 82     LDX  82       csatorna-szám
D3AC  20 25 D1  JSR  D125     file-tipus ellenőrzése
D3AF  D0 03     BNE  D3B4     nem relativ file?
D3B1  4C 20 E1  JMP  E120     byte betöltése a relativ file-ból

D3B4  A5 83     LDA  83       másodlagos cím
D3B6  C9 0F     CMP  #0F      15
D3B8  F0 5A     BEQ  D414     igen, hibacsatorna leolvasása
D3BA  B5 F2     LDA  F2,X
D3BC  29 08     AND  #08      be van állítva a vég-kapcsoló?
D3BE  D0 13     BNE  D3D3     nincs
D3C0  20 25 D1  JSR  D125     file-tipus ellenőrzése
D3C3  C9 07     CMP  #07      közvetlen elérésű file?
D3C5  D0 07     BNE  D3CE     nem
D3C7  A9 89     LDA  #89      a READ- és WRITE-kapcsoló beállítása
D3C9  95 F2     STA  F2,X
D3CB  4C DE D3  JMP  D3DE

D3CE  A9 00     LDA  #00
D3D0  95 F2     STA  F2,X     a READ- és WRITE-kapcsoló törlése
D3D2  60        RTS

D3D3  A5 83     LDA  83       másodlagos cím
D3D5  F0 32     BEQ  D409     nulla, LOAD?
D3D7  20 25 D1  JSR  D125     a file-tipus ellenőrzése
D3DA  C9 04     CMP  #04      relatív file vagy közvetlen elérés?
D3DC  90 22     BCC  D400     nem
D3DE  20 2F D1  JSR  D12F     puffer- és csatorna-szám betöltése
D3E1  B5 99     LDA  99,X     puffer mutató
D3E3  D9 44 02  CMP  0244,Y   azonos a vég-mutatóval?
D3E6  D0 04     BNE  D3EC     nem
```

```
D3E8   A9 00      LDA  #00
D3EA   95 99      STA   99,X          puffer mutató nullán
D3EC   F6 99      INC   99,X          a puffer mutató növelése
D3EE   A1 99      LDA  (99,X)         egy byte beolvasása a pufferból
D3F0   99 3E 02   STA   023E,Y        az output regiszterbe
D3F3   B5 99      LDA   99,X          puffer mutató
D3F5   D9 44 02   CMP   0244,Y        azonos a vég-mutatóval?
D3F8   D0 05      BNE   D3FF          nem
D3FA   A9 81      LDA  #81
D3FC   99 F2 00   STA   00F2,Y        a kapcsolók beállítása
D3FF   60         RTS

D400   20 56 D1   JSR   D156          egy byte beolvasása a pufferból
D403   A6 82      LDX   82            csatorna-szám
D405   9D 3E 02   STA   023E,X        byte az output-regiszterben
D408   60         RTS

D409   AD 54 02   LDA   0254          tartalomjegyzék?
D40C   F0 F2      BEQ   D400          nem
D40E   20 67 ED   JSR   ED67          a tartalomjegyzék sor előállítása
D411   4C 03 D4   JMP   D403

D414   20 E8 D4   JSR   D4E8          a puffer mutató beállítása
D417   C9 D4      CMP  #D4            a puffer előtt hibaüzenetre utal?
D419   D0 18      BNE   D433          nem
D41B   A5 95      LDA   95
D41D   C9 02      CMP  #02
D41F   D0 12      BNE   D433
D421   A9 0D      LDA  #0D            CR
D423   85 85      STA   85            output-regiszterben
D425   20 23 C1   JSR   C123          a hibakapcsolók törlése
D428   A9 00      LDA  #00
D42A   20 C1 E6   JSR   E6C1          az "OK" üzenet előállítása
D42D   C6 A5      DEC   A5            a puffer mutató visszaállítása
D42F   A9 80      LDA  #80            a READ-kapcsoló beállítása
D431   D0 12      BNE   D445
D433   20 37 D1   JSR   D137          a byte beolvasása a pufferból
D436   85 85      STA   85            az output-regiszterbe
D438   D0 09      BNE   D443
D43A   A9 D4      LDA  #D4
D43C   20 C8 D4   JSR   D4C8          a hiba előtti puffer mutató beállítása
D43F   A9 02      LDA  #02
D441   95 9A      STA   9A,X          cím /Hi/
D443   A9 88      LDA  #88            a READ-kapcsoló beállítása
D445   85 F7      STA   F7
D447   A5 85      LDA   85            adatbyte beolvasása
D449   8D 43 02   STA   0243          az output regiszterbe
D44C   60         RTS

******************************        következő blokk olvasása
D44D   20 93 DF   JSR   DF93          puffer-szám beolvasása
D450   0A         ASL   A             kétszer
D451   AA         TAX
D452   A9 00      LDA  #00
D454   95 99      STA   99,X          puffer mutató nullán
D456   A1 99      LDA  (99,X)         első byte beolvasása a pufferból
D458   F0 05      BEQ   D45F          nem folytatóblokk?
D45A   D6 99      DEC   99,X          puffer mutató 1-en
D45C   4C 56 D1   JMP   D156          következő blokk olvasása

D45F   60         RTS

******************************        egy blokk olvasása
D460   A9 80      LDA  #80            olvasás-utasításkód
D462   D0 02      BNE   D466
```

```
****************************        egy blokk irása
D464  A9 90      LDA  #90           irás-utasításkód
D466  05 7F      ORA  7F            egységszám
D468  8D 4D 02   STA  024D          a kód tárolása
D46B  A5 F9      LDA  F9
D46D  20 D3 D6   JSR  D6D3          paraméter a lemezvezérlőn
D470  A6 F9      LDX  F9
D472  4C 93 D5   JMP  D593          utasítás végrehajtása

****************************        puffer lefoglalása és a blokk olvasása
D475  A9 01      LDA  #01
D477  8D 4A 02   STA  024A          soros adatfeldolgozás
D47A  A9 11      LDA  #11           17
D47C  85 83      STA  83            másodlagos cím
D47E  20 46 DC   JSR  DC46          puffer lefoglalása és a blokk olvasása
D481  A9 02      LDA  #02
D483  4C C8 D4   JMP  D4C8          puffer mutató 2-n

****************************        új blokk létrehozása
D486  A9 12      LDA  #12           18
D488  85 83      STA  83            másodlagos cím
D48A  4C DA DC   JMP  DCDA          új blokk létesítése

****************************        tartalomjegyzék-blokk irása
D48D  20 3B DE   JSR  DE3B          a sáv- és szektor-szám beolvasása
D490  A9 01      LDA  #01
D492  85 6F      STA  6F            egy blokk
D494  A5 69      LDA  69            blokk-foglalásnál lépéstávolság tárolása /1Ø/
D496  48         PHA
D497  A9 03      LDA  #03           tartalomjegyzéknél 3!
D499  85 69      STA  69
D49B  20 2D F1   JSR  F12D          egy szabad blokk megkeresése a BAM-ban
D49E  68         PLA
D49F  85 69      STA  69            lépéstávolság visszatöltése
D4A1  A9 00      LDA  #00
D4A3  20 C8 D4   JSR  D4C8          puffer mutató nullán
D4A6  A5 80      LDA  80
D4A8  20 F1 CF   JSR  CFF1          a sáv-szám a pufferban
D4AB  A5 81      LDA  81
D4AD  20 F1 CF   JSR  CFF1          a szektor-szám a pufferban
D4B0  20 C7 D0   JSR  D0C7          a blokk felírása a lemezre
D4B3  20 99 D5   JSR  D599          és ellenőrzése
D4B6  A9 00      LDA  #00
D4B8  20 C8 D4   JSR  D4C8          puffer mutató nullán
D4BB  20 F1 CF   JSR  CFF1          puffer kitöltése nullákkal
D4BE  D0 FB      BNE  D4BB
D4C0  20 F1 CF   JSR  CFF1          nulla, mint láncolt sáv
D4C3  A9 FF      LDA  #FF
D4C5  4C F1 CF   JMP  CFF1          ØFF, mint a byte-ok száma

****************************        puffer mutató beállítása
D4C8  85 6F      STA  6F            a mutató tárolása
D4CA  20 93 DF   JSR  DF93          puffer-szám betöltése
D4CD  0A         ASL  A             kétszer
D4CE  AA         TAX
D4CF  B5 9A      LDA  9A,X          puffer mutató /Hi/
D4D1  85 95      STA  95
D4D3  A5 6F      LDA  6F
D4D5  95 99      STA  99,X          puffer mutató, új érték /Lo/
D4D7  85 94      STA  94
D4D9  60         RTS

****************************        a belső csatorna lezárása
D4DA  A9 11      LDA  #11           17
D4DC  85 83      STA  83
D4DE  20 27 D2   JSR  D227          a csatorna lezárása
D4E1  A9 12      LDA  #12           18
D4E3  85 83      STA  83
D4E5  4C 27 D2   JMP  D227          a csatorna lezárása
```

```
*****************************        puffer mutató beállítása
D4EB  20 93 DF    JSR   DF93         a puffer-szám betöltése
D4EB  0A          ASL   A
D4EC  AA          TAX
D4ED  B5 9A       LDA   9A,X         puffer mutató /Hi/
D4EF  85 95       STA   95
D4F1  B5 99       LDA   99,X         puffer mutató /Lo/
D4F3  85 94       STA   94
D4F5  60          RTS

*****************************        a byte betöltése a pufferból
D4F6  85 71       STA   71           mutató /Lo/
D4F8  20 93 DF    JSR   DF93         a puffer-szám betöltése
D4FB  AA          TAX
D4FC  BD E0 FE    LDA   FEE0,X       puffercím /Hi/
D4FF  85 72       STA   72           mutató /Hi/
D501  A0 00       LDY   #00
D503  B1 71       LDA   (71),Y       egy byte betöltése a pufferból
D505  60          RTS

*****************************        sáv- és szektor-szám ellenőrzése
D506  BD 5B 02    LDA   025B,X       lemezvezérlő utasításkódja
D509  29 01       AND   #01          egységszám
D50B  0D 4D 02    ORA   024D         plusz utasításkód
D50E  48          PHA                tárolása
D50F  86 F9       STX   F9           puffer-szám
D511  8A          TXA
D512  0A          ASL   A            kétszer
D513  AA          TAX
D514  B5 07       LDA   07,X         szektor
D516  8D 4D 02    STA   024D         tárolása
D519  B5 06       LDA   06,X         sáv
D51B  F0 2D       BEQ   D54A         66, "illegal track or sector"
D51D  CD D7 FE    CMP   FED7         36, legmegasabb sáv-szám + 1
D520  B0 28       BCS   D54A         66, "illegal track or sector"
D522  AA          TAX
D523  68          PLA                utasításkód
D524  48          PHA
D525  29 F0       AND   #F0
D527  C9 90       CMP   #90          irás-kód?
D529  D0 4F       BNE   D57A         nem
D52B  68          PLA
D52C  48          PHA
D52D  4A          LSR   A
D52E  B0 05       BCS   D535
D530  AD 01 01    LDA   0101
D533  90 03       BCC   D538
D535  AD 02 01    LDA   0102
D538  F0 05       BEQ   D53F
D53A  CD D5 FE    CMP   FED5         "A",formátum-jelölés
D53D  D0 33       BNE   D572         73, "cbm dos v2.6 1541"
D53F  8A          TXA                sáv-szám
D540  20 4B F2    JSR   F24B         maximális szektor-szám beolvasása
D543  CD 4D 02    CMP   024D         és összehasonlítása a szektor-számmal
D546  F0 02       BEQ   D54A         ha egyenlő: hiba
D548  B0 30       BCS   D57A         kisebb?
D54A  20 52 D5    JSR   D552         sáv- és szektor-szám beolvasása
D54D  A9 66       LDA   #66
D54F  4C 45 E6    JMP   E645         66, "illegal track or sector"

*****************************        sáv- és szektor-szám beolvasása
D552  A5 F9       LDA   F9           puffer-szám
D554  0A          ASL   A            #2
D555  AA          TAX                indexként
D556  B5 06       LDA   06,X
D558  85 80       STA   80           sáv
D55A  B5 07       LDA   07,X
D55C  85 81       STA   81           szektor
D55E  60          RTS
```

```
D55F  A5 80     LDA  80       sáv
D561  F0 EA     BEQ  D54D     ha nulla, akkor hiba
D563  CD D7 FE  CMP  FED7     36, maximális sáv-szám + 1
D566  B0 E5     BCS  D54D     66, "illegal track or sector"
D568  20 4B F2  JSR  F24B     maximális szektor-szám betöltése
D56B  C5 81     CMP  81       szektor
D56D  F0 DE     BEQ  D54D
D56F  90 DC     BCC  D54D     hiba
D571  60        RTS

D572  20 52 D5  JSR  D552     sáv- és szektor-szám betöltése
D575  A9 73     LDA  #73
D577  4C 45 E6  JMP  E645     73, "cbm dos v2.6 1541"

D57A  A6 F9     LDX  F9       puffer-szám
D57C  68        PLA
D57D  8D 4D 02  STA  024D     lemezvezérlő utasításkódja
D580  95 00     STA  00,X     az utasítás-regiszterben
D582  9D 5B 02  STA  025B,X   utasításkód beírása a táblázatba
D585  60        RTS

****************************  blokk olvasása
D586  A9 80     LDA  #80      olvasáskód
D588  D0 02     BNE  D58C

****************************  blokk irása
D58A  A9 90     LDA  #90      íráskód
D58C  05 7F     ORA  7F       egységszám
D58E  A6 F9     LDX  F9       puffer-szám
D590  8D 4D 02  STA  024D
D593  AD 4D 02  LDA  024D     utasításkód
D596  20 0E D5  JSR  D50E     sáv és szektor ellenőrzése

****************************  a végrehajtás ellenőrzése
D599  20 A6 D5  JSR  D5A6     a végrehajtás ellenőrzése
D59C  B0 FB     BCS  D599
D59E  48        PHA           visszajelzés-kód
D59F  A9 00     LDA  #00
D5A1  8D 98 02  STA  0298     a hiba-kapcsoló törlése
D5A4  68        PLA
D5A5  60        RTS
                              az utasításkód /7-es bit/ még a
D5A6  B5 00     LDA  00,X     regiszterben van?
D5A8  30 1A     BMI  D5C4     igen, ha a
D5AA  C9 02     CMP  #02      visszajelzés kisebb 2-nél
D5AC  90 14     BCC  D5C2     hibás végrehajtás
D5AE  C9 08     CMP  #08      ha 8,
D5B0  F0 08     BEQ  D5BA     Write Protect /írásvédelem/
D5B2  C9 0B     CMP  #0B      ha 11,
D5B4  F0 04     BEQ  D5BA     ID mismatch
D5B6  C9 0F     CMP  #0F      15
D5B8  D0 0C     BNE  D5C6
D5BA  2C 98 02  BIT  0298
D5BD  30 03     BMI  D5C2
D5BF  4C 3F D6  JMP  D63F     a hibaüzenet előállítása

D5C2  18        CLC           a végrehajtás befejeződik
D5C3  60        RTS

D5C4  38        SEC           a végrehajtás még nem fejeződik be
D5C5  60        RTS

D5C6  98        TYA
D5C7  48        PHA
D5C8  A5 7F     LDA  7F       egységszám
D5CA  48        PHA
D5CB  BD 5B 02  LDA  025B,X
```

```
D5CE  29 01      AND  #01      egységszám
D5D0  85 7F      STA  7F
D5D2  A8         TAY
D5D3  B9 CA FE   LDA  FECA,Y   az egység bit-mintája
D5D6  8D 6D 02   STA  026D
D5D9  20 A6 D6   JSR  D6A6     olvasási kisérlet
D5DC  C9 02      CMP  #02      visszajelzés
D5DE  B0 03      BCS  D5E3     nem ok?
D5E0  4C 6D D6   JMP  D66D     kész

D5E3  BD 5B 02   LDA  025B,X   utasításkód
D5E6  29 F0      AND  #F0      leválasztása
D5E8  48         PHA
D5E9  C9 90      CMP  #90      íráskód
D5EB  D0 07      BNE  D5F4     nem
D5ED  A5 7F      LDA  7F       egységszám
D5EF  09 B8      ORA  #B8
D5F1  9D 5B 02   STA  025B,X
D5F4  24 6A      BIT  6A
D5F6  70 39      BVS  D631
D5F8  A9 00      LDA  #00
D5FA  8D 99 02   STA  0299     keresés-számláló a sáv mellett
D5FD  8D 9A 02   STA  029A
D600  AC 99 02   LDY  0299     számláló
D603  AD 9A 02   LDA  029A
D606  38         SEC
D607  F9 DB FE   SBC  FEDB,Y   olvasási kisérlet állandói a sáv mellett
D60A  8D 9A 02   STA  029A
D60D  B9 DB FE   LDA  FEDB,Y
D610  20 76 D6   JSR  D676     mágnesfej pozicionálása a sáv mellé
D613  EE 99 02   INC  0299     számláló növelése
D616  20 A6 D6   JSR  D6A6     olvasási kisérlet
D619  C9 02      CMP  #02      visszajelzés
D61B  90 08      BCC  D625     2-nél kisebb, ok?
D61D  AC 99 02   LDY  0299     számláló betöltése
D620  B9 DB FE   LDA  FEDB,Y   az állandók beolvasása
D623  D0 DB      BNE  D600     még nem nulla /táblázat vége/?
D625  AD 9A 02   LDA  029A
D628  20 76 D6   JSR  D676     mágnesfej pozicionálása
D62B  B5 00      LDA  00,X
D62D  C9 02      CMP  #02      visszajelzés
D62F  90 2B      BCC  D65C     ok?
D631  24 6A      BIT  6A
D633  10 0F      BPL  D644
D635  68         PLA           utasításkód
D636  C9 90      CMP  #90      íráshoz?
D638  D0 05      BNE  D63F     nem
D63A  05 7F      ORA  7F       egységszám
D63C  9D 5B 02   STA  025B,X   utasításkód a táblázatban
D63F  B5 00      LDA  00,X     visszajelzés
D641  20 0A E6   JSR  E60A     a hibaüzenet beállítása
D644  68         PLA
D645  2C 98 02   BIT  0298
D648  30 23      BMI  D66D
D64A  48         PHA
D64B  A9 C0      LDA  #C0      fejpozicionálás utasításkódja
D64D  05 7F      ORA  7F       egységszám
D64F  95 00      STA  00,X     az utasítás-regiszterben
D651  B5 00      LDA  00,X
D653  30 FC      BMI  D651     végrehajtás /futtatás/ megvárása
D655  20 A6 D6   JSR  D6A6     végrehajtást mégegyszer megkisérelni!
D658  C9 02      CMP  #02      visszajelzés
D65A  B0 D9      BCS  D635     hibás?
D65C  68         PLA
D65D  C9 90      CMP  #90      írás-utasításkód
D65F  D0 0C      BNE  D66D     nem
D661  05 7F      ORA  7F       egységszám
```

```
D663  9D 5B 02   STA   025B,X    a táblázatban
D666  20 A6 D6   JSR   D6A6      utasítás végrehajtását mégegyszer megismételni!
D669  C9 02      CMP   #02       visszajelzés
D66B  B0 D2      BCS   D63F      hiba?
D66D  68         PLA
D66E  85 7F      STA   7F        az egységszám visszatöltése
D670  68         PLA
D671  A8         TAY
D672  B5 00      LDA   00,X      hibakód
D674  18         CLC             a végrehajtási kapcsoló vége
D675  60         RTS

D676  C9 00      CMP   #00
D678  F0 18      BEQ   D692
D67A  30 0C      BMI   D688
D67C  A0 01      LDY   #01
D67E  20 93 D6   JSR   D693      mágnesfej pozicionáláshoz szükséges adatok
D681  38         SEC             átadása
D682  E9 01      SBC   #01
D684  D0 F6      BNE   D67C
D686  F0 0A      BEQ   D692
D688  A0 FF      LDY   #FF
D68A  20 93 D6   JSR   D693      mágnesfej pozicionáláshoz szükséges adatok
D68D  18         CLC             átadása
D68E  69 01      ADC   #01
D690  D0 F6      BNE   D688
D692  60         RTS

D693  48         PHA
D694  98         TYA
D695  A4 7F      LDY   7F        egységszám
D697  99 FE 02   STA   02FE,Y
D69A  D9 FE 02   CMP   02FE,Y    várakozás a lemezvezérlő visszajelzésére
D69D  F0 FB      BEQ   D69A
D69F  A9 00      LDA   #00
D6A1  99 FE 02   STA   02FE,Y
D6A4  68         PLA
D6A5  60         RTS

D6A6  A5 6A      LDA   6A        ismétlések maximális száma
D6A8  29 3F      AND   #3F
D6AA  A8         TAY
D6AB  AD 6D 02   LDA   026D      LED-bit
D6AE  4D 00 1C   EOR   1C00
D6B1  8D 00 1C   STA   1C00      LED kikapcsolása
D6B4  BD 5B 02   LDA   025B,X    utasítás átadása
D6B7  95 00      STA   00,X      a lemezvezérlőhöz
D6B9  B5 00      LDA   00,X      várakozás a visszajelzésre
D6BB  30 FC      BMI   D6B9
D6BD  C9 02      CMP   #02       ok?
D6BF  90 03      BCC   D6C4      igen
D6C1  88         DEY             számláló csökkentése
D6C2  D0 E7      BNE   D6AB      új kisérlet
D6C4  48         PHA
D6C5  AD 6D 02   LDA   026D
D6C8  0D 00 1C   ORA   1C00      LED KI
D6CB  8D 00 1C   STA   1C00
D6CE  68         PLA
D6CF  60         RTS

****************************     paraméter átadása a lemezvezérlőnek
D6D0  20 93 DF   JSR   DF93      a puffer-szám betöltése
D6D3  0A         ASL   A
D6D4  A8         TAY
D6D5  A5 80      LDA   80        sáv-szám
D6D7  99 06 00   STA   0006,Y    átadása
D6DA  A5 81      LDA   81        szektor-szám
```

```
D6DC  99 07 00   STA   0007,Y    átadása
D6DF  A5 7F      LDA   7F        egységszám
D6E1  0A         ASL   A         2-szer
D6E2  AA         TAX             X után
D6E3  60         RTS

****************************     a file bejegyzése a tartalomjegyzékbe
D6E4  A5 83      LDA   83        másodlagos cím
D6E6  48         PHA
D6E7  A5 82      LDA   82        csatorna-szám
D6E9  48         PHA
D6EA  A5 81      LDA   81        szektor-szám
D6EC  48         PHA
D6ED  A5 80      LDA   80        sáv-szám
D6EF  48         PHA             tárolása
D6F0  A9 11      LDA   #11
D6F2  85 83      STA   83        17 másodlagos cím
D6F4  20 3B DE   JSR   DE3B      sáv- és szektor-szám betöltése
D6F7  AD 4A 02   LDA   024A      file-tipus
D6FA  48         PHA             tárolása
D6FB  A5 E2      LDA   E2        egységszám
D6FD  29 01      AND   #01
D6FF  85 7F      STA   7F        beállítása
D701  A6 F9      LDX   F9        puffer-szám
D703  5D 5B 02   EOR   025B,X
D706  4A         LSR   A
D707  90 0C      BCC   D715      azonos egységszám?
D709  A2 01      LDX   #01
D70B  8E 92 02   STX   0292      mutató a tartalomjegyzékben
D70E  20 AC C5   JSR   C5AC      a tartalomjegyzék betöltése
D711  F0 1D      BEQ   D730      első bejegyzés
D713  D0 28      BNE   D73D      megvan?
D715  AD 91 02   LDA   0291      szektor-szám a tartalomjegyzékben
D718  F0 0C      BEQ   D726      nullával egyenlő
D71A  C5 81      CMP   81        azonos szektor-számok?
D71C  F0 1F      BEQ   D73D      igen
D71E  85 81      STA   81        a szektor-szám tárolása
D720  20 60 D4   JSR   D460      blokk olvasása
D723  4C 3D D7   JMP   D73D

D726  A9 01      LDA   #01
D728  8D 92 02   STA   0292      a mutató 1-en
D72B  20 17 C6   JSR   C617      következő bejegyzés
D72E  D0 0D      BNE   D73D      megvan?
D730  20 8D D4   JSR   D48D      tartalomjegyzék-blokk írása
D733  A5 81      LDA   81        szektor-szám
D735  8D 91 02   STA   0291
D738  A9 02      LDA   #02
D73A  8D 92 02   STA   0292      mutató 2
D73D  AD 92 02   LDA   0292
D740  20 C8 D4   JSR   D4C8      a puffer mutató beállítása
D743  68         PLA
D744  8D 4A 02   STA   024A      file-tipus
D747  C9 04      CMP   #04       relatív file?
D749  D0 02      BNE   D74D      nem
D74B  09 80      ORA   #80       a 7-es bit beállítása
D74D  20 F1 CF   JSR   CFF1      és beírása a pufferbe
D750  68         PLA
D751  8D 80 02   STA   0280      láncolt sáv
D754  20 F1 CF   JSR   CFF1      a pufferben
D757  68         PLA
D758  8D 85 02   STA   0285      láncolt szektor
D75B  20 F1 CF   JSR   CFF1      a pufferben
D75E  20 93 DF   JSR   DF93      a puffer-szám betöltése
D761  A8         TAY
D762  AD 7A 02   LDA   027A      mutató az egységszámra
D765  AA         TAX
```

```
D766  A9 10      LDA  #10       16, file-név hossza
D768  20 6E C6   JSR  C66E      a file-név beírása a pufferbe
D76B  A0 10      LDY  #10
D76D  A9 00      LDA  #00
D76F  91 94      STA  (94),Y    a 16. pozíciótól kezdve nulla!
D771  C8         INY
D772  C0 1B      CPY  #1B       a 27-es pozíció?
D774  90 F9      BCC  D76F      nem
D776  AD 4A 02   LDA  024A      file-tipus
D779  C9 04      CMP  #04       relatív file
D77B  D0 13      BNE  D790      nem
D77D  A0 10      LDY  #10
D77F  AD 59 02   LDA  0259      sáv
D782  91 94      STA  (94),Y
D784  C8         INY
D785  AD 5A 02   LDA  025A      és szektor
D788  91 94      STA  (94),Y    az oldal-szektorokból
D78A  C8         INY
D78B  AD 58 02   LDA  0258      rekordhossz
D78E  91 94      STA  (94),Y
D790  20 64 D4   JSR  D464      blokk írása
D793  68         PLA
D794  85 82      STA  82        csatorna-szám
D796  AA         TAX
D797  68         PLA
D798  85 83      STA  83        másodlagos cím
D79A  AD 91 02   LDA  0291
D79D  85 D8      STA  D8
D79F  9D 60 02   STA  0260,X
D7A2  AD 92 02   LDA  0292
D7A5  85 DD      STA  DD
D7A7  9D 66 02   STA  0266,X
D7AA  AD 4A 02   LDA  024A      file-tipus
D7AD  85 E7      STA  E7
D7AF  A5 7F      LDA  7F        egységszám
D7B1  85 E2      STA  E2
D7B3  60         RTS

******************************  OPEN utasítás, <> 15 másodlagos cím
D7B4  A5 83      LDA  83        másodlagos cím
D7B6  8D 4C 02   STA  024C
D7B9  20 B3 C2   JSR  C2B3      a sorhosszúság, kapcsolók törlése
D7BC  8E 2A 02   STX  022A
D7BF  AE 00 02   LDX  0200      első karakter a pufferből
D7C2  AD 4C 02   LDA  024C      másodlagos cím
D7C5  D0 2C      BNE  D7F3      nem egyenlő Ø-val /LOAD/?
D7C7  E0 2A      CPX  #2A       "*"
D7C9  D0 28      BNE  D7F3
D7CB  A5 7E      LDA  7E        utolsó sáv-szám
D7CD  F0 4D      BEQ  D81C
D7CF  85 80      STA  80        sáv-szám
D7D1  AD 6E 02   LDA  026E      utolsó egységszám
D7D4  85 7F      STA  7F        egységszám
D7D6  85 E2      STA  E2
D7D8  A9 02      LDA  #02
D7DA  85 E7      STA  E7        file-tipus a programban
D7DC  AD 6F 02   LDA  026F      utolsó szektor-szám
D7DF  85 81      STA  81        szektor
D7E1  20 00 C1   JSR  C100      LED kikapcsolása
D7E4  20 46 DC   JSR  DC46      puffer lefoglalása, blokk olvasása
D7E7  A9 04      LDA  #04       file-tipus
D7E9  05 7F      ORA  7F        egységszám
D7EB  A6 82      LDX  82        csatorna-szám
D7ED  99 EC 00   STA  00EC,Y    kapcsoló beállítása
D7F0  4C 94 C1   JMP  C194      kész
```

```
D7F3   E0 24      CPX  #24     "$"
D7F5   D0 1E      BNE  D815    nem
D7F7   AD 4C 02   LDA  024C    másodlagos cím
D7FA   D0 03      BNE  D7FF    nem egyenlő nullával?
D7FC   4C 55 DA   JMP  DA55    OPEN $

D7FF   20 D1 C1   JSR  C1D1    sor vizsgálata
D802   AD 85 FE   LDA  FE85    18, tartalomjegyzék
D805   85 80      STA  80      sáv
D807   A9 00      LDA  #00
D809   85 81      STA  81      0 szektor
D80B   20 46 DC   JSR  DC46    puffer lefoglalása, blokk olvasása
D80E   A5 7F      LDA  7F      egységszám
D810   09 02      ORA  #02
D812   4C EB D7   JMP  D7EB    a fentiek szerint tovább

D815   E0 23      CPX  #23     "#"
D817   D0 12      BNE  D82B
D819   4C 84 CB   JMP  CB84    a közvetlen elérésű file megnyitása

D81C   A9 02      LDA  #02
D81E   8D 96 02   STA  0296    file-tipus
D821   A9 00      LDA  #00
D823   85 7F      STA  7F      0 meghajtó
D825   8D 8E 02   STA  028E
D828   20 42 D0   JSR  D042    a BAM betöltése
D82B   20 E5 C1   JSR  C1E5    a sor elemzése
D82E   D0 04      BNE  D834    sikerült megtalálni a kettőspontot?
D830   A2 00      LDX  #00
D832   F0 0C      BEQ  D840
D834   8A         TXA          sikerült megtalálni a vesszőt?
D835   F0 05      BEQ  D83C    nem
D837   A9 30      LDA  #30
D839   4C C8 C1   JMP  C1C8    30, "syntax error"

D83C   88         DEY
D83D   F0 01      BEQ  D840
D83F   88         DEY
D840   8C 7A 02   STY  027A    mutató az egységszámon
D843   A9 8D      LDA  #8D     shift CR
D845   20 68 C2   JSR  C268    a sor vizsgálata
D848   E8         INX
D849   8E 78 02   STX  0278    vessző-számláló
D84C   20 12 C3   JSR  C312    az egységszám betöltése
D84F   20 CA C3   JSR  C3CA    az egységszám ellenőrzése
D852   20 9D C4   JSR  C49D    bejegyzés megkeresése
D855   A2 00      LDX  #00     alapértelmezések
D857   8E 58 02   STX  0258    rekordhosszúság
D85A   8E 97 02   STX  0297
D85D   8E 4A 02   STX  024A    file-tipus
D860   E8         INX
D861   EC 77 02   CPX  0277    vessző az egyenlőségjel előtt van?
D864   B0 10      BCS  D876    nem
D866   20 09 DA   JSR  DA09    a file-tipus és az üzemmód beolvasása
D869   E8         INX
D86A   EC 77 02   CPX  0277    további vessző?
D86D   B0 07      BCS  D876    nem
D86F   C0 04      CPY  #04
D871   F0 3E      BEQ  D8B1
D873   20 09 DA   JSR  DA09    a file-tipus és az üzemmód betöltése
D876   AE 4C 02   LDX  024C
D879   86 83      STX  83      másodlagos cím
D87B   E0 02      CPX  #02     nagyobb vagy egyenlő, mint 2?
D87D   B0 12      BCS  D891    igen
D87F   8E 97 02   STX  0297    0 vagy 1 /LOAD vagy SAVE/
D882   A9 40      LDA  #40
```

```
D884  8D F9 02   STA   02F9
D887  AD 4A 02   LDA   024A      file-tipus
D88A  D0 1B      BNE   D8A7      nem eltolt /léptetett/
D88C  A9 02      LDA   #02       prg
D88E  8D 4A 02   STA   024A      file-tipus
D891  AD 4A 02   LDA   024A
D894  D0 11      BNE   D8A7
D896  A5 E7      LDA   E7
D898  29 07      AND   #07       a file-tipus beolvasása az utasítás-sorból
D89A  8D 4A 02   STA   024A
D89D  AD 80 02   LDA   0280      sáv-szám
D8A0  D0 05      BNE   D8A7      nem egyenlő nullával?
D8A2  A9 01      LDA   #01
D8A4  8D 4A 02   STA   024A      szekvenciális /soros/ file-tipus
D8A7  AD 97 02   LDA   0297      üzemmód
D8AA  C9 01      CMP   #01       "W"
D8AC  F0 18      BEQ   D8C6      igen
D8AE  4C 40 D9   JMP   D940

D8B1  BC 7A 02   LDY   027A,X    mutató a második vessző után
D8B4  B9 00 02   LDA   0200,Y    betöltés
D8B7  8D 58 02   STA   0258      rekordhosszúság
D8BA  AD 80 02   LDA   0280      sáv-szám
D8BD  D0 B7      BNE   D876
D8BF  A9 01      LDA   #01       "W"
D8C1  8D 97 02   STA   0297      mint üzemmód
D8C4  D0 B0      BNE   D876
D8C6  A5 E7      LDA   E7        file-tipus
D8C8  29 80      AND   #80       a joker-kapcsoló leválasztása
D8CA  AA         TAX
D8CB  D0 14      BNE   D8E1      joker a névben
D8CD  A9 20      LDA   #20
D8CF  24 E7      BIT   E7        le volt zárva a file?
D8D1  F0 06      BEQ   D8D9      igen
D8D3  20 B6 C8   JSR   C8B6      Ø byte a pufferben és blokk írása
D8D6  4C E3 D9   JMP   D9E3      az oldal-szektor bejegyzése, kész

D8D9  AD 80 02   LDA   0280      első blokk sáv-száma
D8DC  D0 03      BNE   D8E1
D8DE  4C E3 D9   JMP   D9E3      oldal-szektor bejegyzése

D8E1  AD 00 02   LDA   0200      első karakter az input-pufferból
D8E4  C9 40      CMP   #40       ' ' felső vessző?
D8E6  F0 0D      BEQ   D8F5      igen
D8E8  8A         TXA
D8E9  D0 05      BNE   D8F0      be van már állítva a joker?
D8EB  A9 63      LDA   #63       63, "file exists"
D8ED  4C C8 C1   JMP   C1C8

D8F0  A9 33      LDA   #33
D8F2  4C C8 C1   JMP   C1C8      33, "syntax error"

******************************   egy file megnyitása felülírással
D8F5  A5 E7      LDA   E7        file-tipus
D8F7  29 07      AND   #07       leválasztás
D8F9  CD 4A 02   CMP   024A
D8FC  D0 67      BNE   D965      eltérőek a file-tipusok?
D8FE  C9 04      CMP   #04       relatív file?
D900  F0 63      BEQ   D965      64, "file type mismatch"
D902  20 DA DC   JSR   DCDA      új szektor tárolása
D905  A5 82      LDA   82
D907  8D 70 02   STA   0270      csatorna-szám tárolása
D90A  A9 11      LDA   #11
D90C  85 83      STA   83        17-es csatorna
D90E  20 EB D0   JSR   D0EB      olvasás-csatorna megnyitása
D911  AD 94 02   LDA   0294
D914  20 C8 D4   JSR   D4C8      tartalomjegyzék puffermutatójának beállítása
```

```
D917  A0 00     LDY  #00
D919  B1 94     LDA  (94),Y     file-tipus
D91B  09 20     ORA  #20        az 5-ös bit beállítása, file megnyitása
D91D  91 94     STA  (94),Y
D91F  A0 1A     LDY  #1A
D921  A5 80     LDA  80         sáv
D923  91 94     STA  (94),Y
D925  C8        INY
D926  A5 81     LDA  81         és szektor
D928  91 94     STA  (94),Y     "felső vessző" karakteres megnyitáskor
D92A  AE 70 02  LDX  0270       csatorna-szám
D92D  A5 D8     LDA  D8
D92F  9D 60 02  STA  0260,X     mutató a tartalomjegyzék-blokkban
D932  A5 DD     LDA  DD
D934  9D 66 02  STA  0266,X
D937  20 3B DE  JSR  DE3B       sáv- és szektor-szám betöltése
D93A  20 64 D4  JSR  D464       blokk írása
D93D  4C EF D9  JMP  D9EF       sáv-, szektor- és egységszám előkészitése

D940  AD 80 02  LDA  0280       első sáv-szám
D943  D0 05     BNE  D94A       a file nincs törölve?
D945  A9 62     LDA  #62
D947  4C C8 C1  JMP  C1C8       62, "file not found"

D94A  AD 97 02  LDA  0297       üzemmód
D94D  C9 03     CMP  #03        "M"
D94F  F0 0B     BEQ  D95C
D951  A9 20     LDA  #20        5-ös bit
D953  24 E7     BIT  E7         tesztelése a file-tipusban, ha
D955  F0 05     BEQ  D95C       nincs beállítva, ok
D957  A9 60     LDA  #60
D959  4C C8 C1  JMP  C1C8       6Ø, "write file open"

D95C  A5 E7     LDA  E7
D95E  29 07     AND  #07        file-tipus
D960  CD 4A 02  CMP  024A       megegyezik az utasításból származó tipussal?
D963  F0 05     BEQ  D96A       igen
D965  A9 64     LDA  #64
D967  4C C8 C1  JMP  C1C8       64, "file type mismatch"

D96A  A0 00     LDY  #00
D96C  8C 79 02  STY  0279
D96F  AE 97 02  LDX  0297       üzemmód
D972  E0 02     CPX  #02        "A", Append
D974  D0 1A     BNE  D990       nem
D976  C9 04     CMP  #04        relatív file?
D978  F0 EB     BEQ  D965       ha igen, hiba
D97A  B1 94     LDA  (94),Y
D97C  29 4F     AND  #4F        a 4, 5 és 7 bit törlése
D97E  91 94     STA  (94),Y     tárolása
D980  A5 83     LDA  83         csatorna-szám tárolása
D982  48        PHA
D983  A9 11     LDA  #11
D985  85 83     STA  83         17-es csatorna
D987  20 3B DE  JSR  DE3B       a sáv- és szektor-szám beolvasása
D98A  20 64 D4  JSR  D464       blokk irása
D98D  68        PLA
D98E  85 83     STA  83         a csatorna-szám visszatöltése
D990  20 A0 D9  JSR  D9A0       oldal-szektor paraméter
D993  AD 97 02  LDA  0297       üzemmód
D996  C9 02     CMP  #02        "A" Append
D998  D0 55     BNE  D9EF       nem
D99A  20 2A DA  JSR  DA2A       append előkészítése
D99D  4C 94 C1  JMP  C194       kész, lemezstátus előkészítése
```

```
D9A0  A0 13     LDY  #13
D9A2  B1 94     LDA  (94),Y      sáv
D9A4  8D 59 02  STA  0259
D9A7  C8        INY
D9A8  B1 94     LDA  (94),Y      és szektor az első oldal-szektor-blokkból
D9AA  8D 5A 02  STA  025A
D9AD  C8        INY
D9AE  B1 94     LDA  (94),Y      rekordhosszúság
D9B0  AE 58 02  LDX  0258        utolsó rekordhosszúság
D9B3  8D 58 02  STA  0258
D9B6  8A        TXA
D9B7  F0 0A     BEQ  D9C3        utolsó rekordhosszúság nulla
D9B9  CD 58 02  CMP  0258        azonos a rekordhosszúság?
D9BC  F0 05     BEQ  D9C3        igen
D9BE  A9 50     LDA  #50
D9C0  20 C8 C1  JSR  C1C8        50, "record not present"
D9C3  AE 79 02  LDX  0279
D9C6  BD 80 02  LDA  0280,X
D9C9  85 80     STA  80          sáv
D9CB  BD 85 02  LDA  0285,X
D9CE  85 81     STA  81          szektor
D9D0  20 46 DC  JSR  DC46
D9D3  A4 82     LDY  82
D9D5  AE 79 02  LDX  0279
D9D8  B5 D8     LDA  D8,X
D9DA  99 60 02  STA  0260,Y
D9DD  B5 DD     LDA  DD,X
D9DF  99 66 02  STA  0266,Y
D9E2  60        RTS

D9E3  A5 E2     LDA  E2          egységszám
D9E5  29 01     AND  #01         leválasztása
D9E7  85 7F     STA  7F          egységszám
D9E9  20 DA DC  JSR  DCDA
D9EC  20 E4 D6  JSR  D6E4        a file bejegyzése
D9EF  A5 83     LDA  83          csatorna-szám
D9F1  C9 02     CMP  #02
D9F3  B0 11     BCS  DA06        nagyobb vagy egyenlő mint 2?
D9F5  20 3E DE  JSR  DE3E        sáv- és szektor-szám beolvasása
D9F8  A5 80     LDA  80
D9FA  85 7E     STA  7E
D9FC  A5 7F     LDA  7F          egységszám
D9FE  8D 6E 02  STA  026E
DA01  A5 81     LDA  81          szektor
DA03  8D 6F 02  STA  026F
DA06  4C 99 C1  JMP  C199

*****************************    a file-tipus és az üzemmód ellenőrzése
DA09  BC 7A 02  LDY  027A,X      mutató az utasítás-sorban
DA0C  B9 00 02  LDA  0200,Y      karakter beolvasása a sorból
DA0F  A0 04     LDY  #04
DA11  88        DEY
DA12  30 08     BMI  DA1C
DA14  D9 B2 FE  CMP  FEB2,Y      "R", "W", "A", "M" üzemmód
DA17  D0 F8     BNE  DA11
DA19  8C 97 02  STY  0297        tárolása
DA1C  A0 05     LDY  #05
DA1E  88        DEY
DA1F  30 08     BMI  DA29
DA21  D9 B6 FE  CMP  FEB6,Y      "D", "S", "P", "U", "L" file-tipus
DA24  D0 F8     BNE  DA1E
DA26  8C 4A 02  STY  024A        tárolása
DA29  60        RTS

*****************************    előkészités az append számára
DA2A  20 39 CA  JSR  CA39        olvasás-csatorna megnyitása, egy byte olvasása
DA2D  A9 80     LDA  #80
```

```
DA2F  20 A6 DD   JSR  DDA6      utolsó byte?
DA32  F0 F6      BEQ  DA2A      nem
DA34  20 95 DE   JSR  DE95      a sáv- és a szektor-szám olvasása
DA37  A6 81      LDX  81        szektor-szám
DA39  E8         INX
DA3A  8A         TXA
DA3B  D0 05      BNE  DA42      nem $FF?
DA3D  20 A3 D1   JSR  D1A3      puffer lezárása, blokk írása
DA40  A9 02      LDA  #02
DA42  20 C8 D4   JSR  D4C8      puffer mutató 2-n
DA45  A6 82      LDX  82        csatorna-szám
DA47  A9 01      LDA  #01
DA49  95 F2      STA  F2,X      írás-kapcsoló beállítása
DA4B  A9 80      LDA  #80
DA4D  05 82      ORA  82
DA4F  A6 83      LDX  83
DA51  9D 2B 02   STA  022B,X    csatorna-szám a táblázatban
DA54  60         RTS

******************************  OPEN "$"
DA55  A9 0C      LDA  #0C       12 utasítás-szám
DA57  8D 2A 02   STA  022A
DA5A  A9 00      LDA  #00
DA5C  AE 74 02   LDX  0274
DA5F  CA         DEX
DA60  F0 0B      BEQ  DA6D
DA62  CA         DEX
DA63  D0 21      BNE  DA86
DA65  AD 01 02   LDA  0201      második karakter
DA68  20 BD C3   JSR  C3BD      az egységszám beolvasása
DA6B  30 19      BMI  DA86      nem egyértelmű?
DA6D  85 E2      STA  E2
DA6F  EE 77 02   INC  0277
DA72  EE 78 02   INC  0278
DA75  EE 7A 02   INC  027A
DA78  A9 80      LDA  #80
DA7A  85 E7      STA  E7        a joker-kapcsoló beállítása
DA7C  A9 2A      LDA  #2A       "*"
DA7E  8D 00 02   STA  0200      file-névként az utasítás-pufferben
DA81  8D 01 02   STA  0201
DA84  D0 18      BNE  DA9E      feltétlen ugrás
DA86  20 E5 C1   JSR  C1E5      input-sor tesztelése ":"-ig
DA89  D0 05      BNE  DA90      ":" ?
DA8B  20 DC C2   JSR  C2DC      kapcsolók törlése
DA8E  A0 03      LDY  #03
DA90  88         DEY
DA91  88         DEY
DA92  8C 7A 02   STY  027A
DA95  20 00 C2   JSR  C200      a sor elemzése
DA98  20 98 C3   JSR  C398      a file-tipus meghatározása
DA9B  20 20 C3   JSR  C320
DA9E  20 CA C3   JSR  C3CA      szükség esetén az egység inicializálása
DAA1  20 B7 C7   JSR  C7B7      a lemezcím előkészítése
DAA4  20 9D C4   JSR  C49D      tartalomjegyzék betöltés
DAA7  20 9E EC   JSR  EC9E
DAAA  20 37 D1   JSR  D137      egy byte a pufferból
DAAD  A6 82      LDX  82        csatorna-szám
DAAF  9D 3E 02   STA  023E,X    byte az output-regiszterben
DAB2  A5 7F      LDA  7F        egységszám
DAB4  8D 8E 02   STA  028E      tárolás utolsó egységszámként
DAB7  09 04      ORA  #04
DAB9  95 EC      STA  EC,X      prg kapcsoló
DABB  A9 00      LDA  #00
DABD  85 A3      STA  A3        a mutató visszaállítása az input-pufferben!
DABF  60         RTS
```

```
*****************************          CLOSE-rutin
DAC0  A9 00      LDA  #00
DAC2  8D F9 02   STA  02F9
DAC5  A5 83      LDA  83               másodlagos cím
DAC7  D0 0B      BNE  DAD4             nem egyenlő nullával?
DAC9  A9 00      LDA  #00              Ø másodlagos cím, LOAD
DACB  8D 54 02   STA  0254
DACE  20 27 D2   JSR  D227             csatorna lezárása
DAD1  4C DA D4   JMP  D4DA             a 17 és 18 belső csatorna lezárása

DAD4  C9 0F      CMP  #0F              15
DAD6  F0 14      BEQ  DAEC             ha igen, minden csatornát lezárni!
DAD8  20 02 DB   JSR  DB02             file-t lezárni!
DADB  A5 83      LDA  83               másodlagos cím
DADD  C9 02      CMP  #02
DADF  90 F0      BCC  DAD1             2-nél kisebb?
DAE1  AD 6C 02   LDA  026C
DAE4  D0 03      BNE  DAE9
DAE6  4C 94 C1   JMP  C194             lezárás

DAE9  4C AD C1   JMP  C1AD

DAEC  A9 0E      LDA  #0E              14
DAEE  85 83      STA  83               másodlagos cím
DAF0  20 02 DB   JSR  DB02             file-t lezárni!
DAF3  C6 83      DEC  83               következő másodlagos cím
DAF5  10 F9      BPL  DAF0
DAF7  AD 6C 02   LDA  026C
DAFA  D0 03      BNE  DAFF
DAFC  4C 94 C1   JMP  C194             lezárás

DAFF  4C AD C1   JMP  C1AD

*****************************          file lezárása
DB02  A6 83      LDX  83               másodlagos cím
DB04  BD 2B 02   LDA  022B,X           csatorna-szám beolvasása
DB07  C9 FF      CMP  #FF              nincs csatorna hozzárendelve?
DB09  D0 01      BNE  DB0C
DB0B  60         RTS                   ha nincs, kész

DB0C  29 0F      AND  #0F              csatorna-szám leválasztása
DB0E  85 82      STA  82
DB10  20 25 D1   JSR  D125             file-tipus ellenőrzése
DB13  C9 07      CMP  #07              közvetlen elérés?
DB15  F0 0F      BEQ  DB26             igen
DB17  C9 04      CMP  #04              relatív file?
DB19  F0 11      BEQ  DB2C             igen
DB1B  20 07 D1   JSR  D107             írás-csatorna megnyitása
DB1E  B0 09      BCS  DB29             nincs file íráshoz?
DB20  20 62 DB   JSR  DB62             utolsó blokk írása
DB23  20 A5 DB   JSR  DBA5             bejegyzés a tartalomjegyzékben, egy blokk írása
DB26  20 F4 EE   JSR  EEF4             BAM írása
DB29  4C 27 D2   JMP  D227             csatorna lezárása

DB2C  20 F1 DD   JSR  DDF1             puffer-szám beolvasása és blokk írása
DB2F  20 1E CF   JSR  CF1E             puffer váltás
DB32  20 CB E1   JSR  E1CB             az utolsó oldal-szektor beolvasása
DB35  A6 D5      LDX  D5               oldal-szektor-szám
DB37  86 73      STX  73
DB39  E6 73      INC  73
DB3B  A9 00      LDA  #00
DB3D  85 70      STA  70
DB3F  85 71      STA  71
DB41  A5 D6      LDA  D6
DB43  38         SEC
DB44  E9 0E      SBC  #0E
DB46  85 72      STA  72
```

```
DB48   20 51 DF    JSR   DF51        a file blokk-számának kiszámítása
DB4B   A6 82       LDX   82          csatorna-szám
DB4D   A5 70       LDA   70
DB4F   95 B5       STA   B5,X        rekord-szám /Lo/
DB51   A5 71       LDA   71
DB53   95 BB       STA   BB,X        rekord-szám /Hi/
DB55   A9 40       LDA   #40
DB57   20 A6 DD    JSR   DDA6        be van állítva a 6-os bit?
DB5A   F0 03       BEQ   DB5F        nincs
DB5C   20 A5 DB    JSR   DBA5        bejegyzés!
DB5F   4C 27 D2    JMP   D227        csatorna lezárása

****************************         utolsó blokk írása
DB62   A6 82       LDX   82          csatorna-szám
DB64   B5 B5       LDA   B5,X        rekord-szám /Lo/
DB66   15 BB       ORA   BB,X        rekord-szám /Hi/
DB68   D0 0C       BNE   DB76        nem egyenlő nullával?
DB6A   20 E8 D4    JSR   D4E8        a puffer mutató beállítása
DB6D   C9 02       CMP   #02
DB6F   D0 05       BNE   DB76        nem egyenlő 2-vel
DB71   A9 0D       LDA   #0D         CR
DB73   20 F1 CF    JSR   CFF1        a pufferben
DB76   20 E8 D4    JSR   D4E8        a puffer mutató beállítása
DB79   C9 02       CMP   #02         most egyenlő 2-vel?
DB7B   D0 0F       BNE   DB8C        nem
DB7D   20 1E CF    JSR   CF1E        puffer váltás
DB80   A6 82       LDX   82          csatorna-szám
DB82   B5 B5       LDA   B5,X        rekord-szám /Lo/
DB84   D0 02       BNE   DB88
DB86   D6 BB       DEC   BB,X        rekord-szám /Hi/
DB88   D6 B5       DEC   B5,X
DB8A   A9 00       LDA   #00
DB8C   38          SEC
DB8D   E9 01       SBC   #01         a mutató a végére!
DB8F   48          PHA
DB90   A9 00       LDA   #00
DB92   20 C8 D4    JSR   D4C8        puffer mutató nullán
DB95   20 F1 CF    JSR   CFF1        nulla a pufferbe!
DB98   68          PLA               második byte =a végére utaló mutatóval
DB99   20 F1 CF    JSR   CFF1        írás a pufferbe!
DB9C   20 C7 D0    JSR   D0C7        blokk felírása a lemezre
DB9F   20 99 D5    JSR   D599        ellenőrzés
DBA2   4C 1E CF    JMP   CF1E        puffer váltás

****************************         file-bejegyzés
DBA5   A6 82       LDX   82          csatorna tárolása
DBA7   8E 70 02    STX   0270
DBAA   A5 83       LDA   83          másodlagos cím tárolása
DBAC   48          PHA
DBAD   BD 60 02    LDA   0260,X      szektor-szám beállítása
DBB0   85 81       STA   81          /tartalomjegyzék/
DBB2   BD 66 02    LDA   0266,X      mutató a tartalomjegyzékben
DBB5   8D 94 02    STA   0294
DBB8   B5 EC       LDA   EC,X
DBBA   29 01       AND   #01
DBBC   85 7F       STA   7F          egységszám
DBBE   AD 85 FE    LDA   FE85        18, tartalomjegyzék sáv
DBC1   85 80       STA   80          beállítása
DBC3   20 93 DF    JSR   DF93        puffer-szám növelése
DBC6   48          PHA
DBC7   85 F9       STA   F9
DBC9   20 60 D4    JSR   D460        tartalomjegyzék-blokk olvasása
DBCC   A0 00       LDY   #00
DBCE   BD E0 FE    LDA   FEE0,X      puffer-cím
DBD1   85 87       STA   87
DBD3   AD 94 02    LDA   0294        puffer mutató
DBD6   85 86       STA   86
```

```
DBD8  B1 86     LDA  (86),Y    file-tipus
DBDA  29 20     AND  #20       lezárt file?
DBDC  F0 43     BEQ  DC21      igen
DBDE  20 25 D1  JSR  D125      file-tipus ellenőrzése
DBE1  C9 04     CMP  #04       relatív file?
DBE3  F0 44     BEQ  DC29      igen
DBE5  B1 86     LDA  (86),Y
DBE7  29 8F     AND  #8F       a 4. 5. és 6. bit törlése
DBE9  91 86     STA  (86),Y    a file-tipusban
DBEB  C8        INY
DBEC  B1 86     LDA  (86),Y    sáv-szám
DBEE  85 80     STA  80
DBF0  84 71     STY  71
DBF2  A0 1B     LDY  #1B
DBF4  B1 86     LDA  (86),Y    felülíráskor a file szektor-számának
DBF6  48        PHA            tárolása
DBF7  88        DEY
DBF8  B1 86     LDA  (86),Y
DBFA  D0 0A     BNE  DC06
DBFC  85 80     STA  80        a sáv-szám beállítása
DBFE  68        PLA
DBFF  85 81     STA  81        szektor-szám
DC01  A9 67     LDA  #67
DC03  20 45 E6  JSR  E645      67, "illegal track or sector"
DC06  48        PHA
DC07  A9 00     LDA  #00
DC09  91 86     STA  (86),Y    sáv-szám
DC0B  C8        INY
DC0C  91 86     STA  (86),Y    és a helyettesítő file szektor-szám törlése
DC0E  68        PLA
DC0F  A4 71     LDY  71
DC11  91 86     STA  (86),Y
DC13  C8        INY            új file sáv- és szektor-számának beállítása
DC14  B1 86     LDA  (86),Y
DC16  85 81     STA  81
DC18  68        PLA
DC19  91 86     STA  (86),Y
DC1B  20 7D C8  JSR  C87D      korábbi file törlése
DC1E  4C 29 DC  JMP  DC29

DC21  B1 86     LDA  (86),Y    file-tipus beolvasása
DC23  29 0F     AND  #0F       a Ø-3. bitek leválasztása
DC25  09 80     ORA  #80
DC27  91 86     STA  (86),Y
DC29  AE 70 02  LDX  0270      csatorna-szám
DC2C  A0 1C     LDY  #1C
DC2E  B5 B5     LDA  B5,X      blokk-szám /Lo/
DC30  91 86     STA  (86),Y
DC32  C8        INY
DC33  B5 BB     LDA  BB,X      blokk-szám /Hi/
DC35  91 86     STA  (86),Y
DC37  68        PLA            puffer-szám
DC38  AA        TAX
DC39  A9 90     LDA  #90       "IRÁS"-kód
DC3B  05 7F     ORA  7F        egységszám
DC3D  20 90 D5  JSR  D590      blokk írása
DC40  68        PLA
DC41  85 83     STA  83        másodlagos cím
DC43  4C 07 D1  JMP  D107      írás-csatorna megnyitása

****************************** blokk olvasása, a puffer lefoglalása
DC46  A9 01     LDA  #01
DC48  20 E2 D1  JSR  D1E2      írás-csatorna és -puffer keresése
DC4B  20 B6 DC  JSR  DCB6      a mutató beállítása
DC4E  AD 4A 02  LDA  024A      file-tipus
DC51  48        PHA            tárolása
DC52  0A        ASL  A
```

```
DC53   05 7F      ORA   7F        egységszám
DC55   95 EC      STA   EC,X
DC57   20 9B D0   JSR   D09B      a blokk olvasása
DC5A   A6 82      LDX   82        csatorna-szám
DC5C   A5 80      LDA   80        sáv
DC5E   D0 05      BNE   DC65      láncolt sáv?
DC60   A5 81      LDA   81        szektor
DC62   9D 44 02   STA   0244,X    mint vég-mutató
DC65   68         PLA             file-tipus
DC66   C9 04      CMP   #04       relatív file?
DC68   D0 3F      BNE   DCA9      nem
DC6A   A4 83      LDY   83        másodlagos cím
DC6C   B9 2B 02   LDA   022B,Y    csatorna-szám
DC6F   09 40      ORA   #40
DC71   99 2B 02   STA   022B,Y    a READ- és a WRITE-kapcsoló beállítása
DC74   AD 58 02   LDA   0258      rekordhosszúság
DC77   95 C7      STA   C7,X
DC79   20 8E D2   JSR   D28E      az oldal-szektor puffer megkeresése
DC7C   10 03      BPL   DC81      megvan?
DC7E   4C 0F D2   JMP   D20F      70, "no channel"

DC81   A6 82      LDX   82        csatorna-szám
DC83   95 CD      STA   CD,X
DC85   AC 59 02   LDY   0259
DC88   84 80      STY   80        oldal-szektor sávja
DC8A   AC 5A 02   LDY   025A
DC8D   84 81      STY   81        oldal-szektor szektora
DC8F   20 D3 D6   JSR   D6D3      paraméter átadása a lemezvezérlőnek
DC92   20 73 DE   JSR   DE73      blokk olvasása
DC95   20 99 D5   JSR   D599      és ellenőrzése
DC98   A6 82      LDX   82        csatorna-szám
DC9A   A9 02      LDA   #02
DC9C   95 C1      STA   C1,X      írás-mutató
DC9E   A9 00      LDA   #00
DCA0   20 C8 D4   JSR   D4C8      puffer mutató nullán
DCA3   20 53 E1   JSR   E153      a következő rekord megkeresése
DCA6   4C 3E DE   JMP   DE3E      a sáv és a szektor-szám

DCA9   20 56 D1   JSR   D156      byte a pufferból
DCAC   A6 82      LDX   82        csatorna-szám
DCAE   9D 3E 02   STA   023E,X    byte az output-regiszterbe
DCB1   A9 88      LDA   #88       a READ-kapcsoló beállítása
DCB3   95 F2      STA   F2,X
DCB5   60         RTS

******************************    a mutató visszaállítása
DCB6   A6 82      LDX   82        csatorna-szám
DCB8   B5 A7      LDA   A7,X      puffer-szám
DCBA   0A         ASL   A         2-szer
DCBB   A8         TAY
DCBC   A9 02      LDA   #02
DCBE   99 99 00   STA   0099,Y    puffer mutató /Lo/
DCC1   B5 AE      LDA   AE,X
DCC3   09 80      ORA   #80       a 7-es bit beállítása, puffer nem foglalt
DCC5   95 AE      STA   AE,X
DCC7   0A         ASL   A
DCC8   A8         TAY
DCC9   A9 02      LDA   #02
DCCB   99 99 00   STA   0099,Y    puffer mutató /Lo/
DCCE   A9 00      LDA   #00
DCD0   95 B5      STA   B5,X      blokkok száma /Lo/
DCD2   95 BB      STA   BB,X      blokkok száma /Hi/
DCD4   A9 00      LDA   #00
DCD6   9D 44 02   STA   0244,X    vég-mutató
DCD9   60         RTS
```

```
******************************    új blokk behelyezése
DCDA   20 A9 F1   JSR   F1A9      szabad szektor a BAM-ban
DCDD   A9 01      LDA   #01
DCDF   20 DF D1   JSR   D1DF      csatorna megnyitása, puffer lefoglalása
DCE2   20 D0 D6   JSR   D6D0      paraméter átadása a lemezvezérlőnek
DCE5   20 B6 DC   JSR   DCB6      a mutató visszaállítása
DCE8   A6 82      LDX   82        csatorna-szám
DCEA   AD 4A 02   LDA   024A      file-tipus
DCED   48         PHA
DCEE   0A         ASL   A
DCEF   05 7F      ORA   7F        egységszám
DCF1   95 EC      STA   EC,X      tárolása
DCF3   68         PLA
DCF4   C9 04      CMP   #04       relatív file?
DCF6   F0 05      BEQ   DCFD      igen
DCF8   A9 01      LDA   #01
DCFA   95 F2      STA   F2,X      a WRITE-kapcsoló beállítása
DCFC   60         RTS

DCFD   A4 83      LDY   83        másodlagos cím
DCFF   B9 2B 02   LDA   022B,Y    csatorna-szám a táblázatban
DD02   29 3F      AND   #3F       a legfelső két bit törlése
DD04   09 40      ORA   #40       6-os bit beállítása
DD06   99 2B 02   STA   022B,Y    olvasás- és írás-kapcsoló
DD09   AD 58 02   LDA   0258      rekordhosszúság
DD0C   95 C7      STA   C7,X      a táblázatban
DD0E   20 8E D2   JSR   D28E      puffer keresése és lefoglalása
DD11   10 03      BPL   DD16      megvan?
DD13   4C 0F D2   JMP   D20F      70, "no channel"

DD16   A6 82      LDX   82        csatorna-szám
DD18   95 CD      STA   CD,X      oldal-szektor puffer-száma
DD1A   20 C1 DE   JSR   DEC1      puffer törlése
DD1D   20 1E F1   JSR   F11E      szabad blokk keresése a BAM-ban
DD20   A5 80      LDA   80        sáv
DD22   8D 59 02   STA   0259      az első oldal-szektorhoz
DD25   A5 81      LDA   81        szektor
DD27   8D 5A 02   STA   025A      az oldal-szektorhoz
DD2A   A6 82      LDX   82        csatorna-szám
DD2C   B5 CD      LDA   CD,X      puffer-szám
DD2E   20 D3 D6   JSR   D6D3      paraméter átadása a lemezvezérlőnek
DD31   A9 00      LDA   #00
DD33   20 E9 DE   JSR   DEE9      puffer mutató nullán
DD36   A9 00      LDA   #00
DD38   20 8D DD   JSR   DD8D      nulla, sávmutatóként a pufferben
DD3B   A9 11      LDA   #11       17
DD3D   20 8D DD   JSR   DD8D      mint végmutató a pufferben
DD40   A9 00      LDA   #00       nulla
DD42   20 8D DD   JSR   DD8D      mint oldal-szektor-szám a pufferben
DD45   AD 58 02   LDA   0258      rekordhosszúság
DD48   20 8D DD   JSR   DD8D      a pufferben
DD4B   A5 80      LDA   80        a blokk sáv-száma
DD4D   20 8D DD   JSR   DD8D      a pufferben
DD50   A5 81      LDA   81        szektor-szám
DD52   20 8D DD   JSR   DD8D      a pufferben
DD55   A9 10      LDA   #10       16
DD57   20 E9 DE   JSR   DEE9      puffer mutató 16-on
DD5A   20 3E DE   JSR   DE3E      a sáv- és a szektor-szám beolvasása
DD5D   A5 80      LDA   80        első adatblokk sáv-száma
DD5F   20 8D DD   JSR   DD8D      a pufferben
DD62   A5 81      LDA   81        első adatblokk szektor-száma
DD64   20 8D DD   JSR   DD8D      a pufferben
DD67   20 6C DE   JSR   DE6C      blokk felírása a lemezre
DD6A   20 99 D5   JSR   D599      és ellenőrzése
DD6D   A9 02      LDA   #02
DD6F   20 C8 D4   JSR   D4C8      puffer mutató 2-n
DD72   A6 82      LDX   82        csatorna-szám
```

```
DD74  38        SEC
DD75  A9 00     LDA   #00
DD77  F5 C7     SBC   C7,X      rekordhosszúság
DD79  95 C1     STA   C1,X      írásmutató
DD7B  20 E2 E2  JSR   E2E2      puffer törlése
DD7E  20 19 DE  JSR   DE19      baloldali byte beírása a pufferbe
DD81  20 5E DE  JSR   DE5E      blokk felírása a lemezre
DD84  20 99 D5  JSR   D599      ellenőrzés
DD87  20 F4 EE  JSR   EEF4      BAM
DD8A  4C 98 DC  JMP   DC98      kész

****************************    egy byte felírása az oldal-szektor-blokkba
DD8D  48        PHA             byte tárolása
DD8E  A6 82     LDX   82        csatorna-szám
DD90  B5 CD     LDA   CD,X      oldal-szektor pufferszáma
DD92  4C FD CF  JMP   CFFD      byte a pufferbe

****************************    manipulálás a kapcsolókkal
DD95  90 06     BCC   DD9D
DD97  A6 82     LDX   82        csatorna-szám
DD99  15 EC     ORA   EC,X      kapcsoló beállítása
DD9B  D0 06     BNE   DDA3
DD9D  A6 82     LDX   82        csatorna-szám
DD9F  49 FF     EOR   #FF
DDA1  35 EC     AND   EC,X      a kapcsoló törlése
DDA3  95 EC     STA   EC,X
DDA5  60        RTS

DDA6  A6 82     LDX   82        csatorna-szám
DDA8  35 EC     AND   EC,X      a kapcsoló vizsgálata
DDAA  60        RTS

****************************    írás-utasításkód ellenőrzése
DDAB  20 93 DF  JSR   DF93      a puffer-szám olvasása
DDAE  AA        TAX
DDAF  BD 5B 02  LDA   025B,X
DDB2  29 F0     AND   #F0       az utasításkód leválasztása
DDB4  C9 90     CMP   #90       íráskód?
DDB6  60        RTS

****************************
DDB7  A2 00     LDX   #00
DDB9  86 71     STX   71        másodlagos címek számlálója
DDBB  BD 2B 02  LDA   022B,X    csatorna-szám betöltése a táblázatból
DDBE  C9 FF     CMP   #FF
DDC0  D0 08     BNE   DDCA      a file nyitva van?
DDC2  A6 71     LDX   71
DDC4  E8        INX             a számláló növelése
DDC5  E0 10     CPX   #10       kisebb 16-nál?
DDC7  90 F0     BCC   DDB9
DDC9  60        RTS

DDCA  86 71     STX   71
DDCC  29 3F     AND   #3F       a csatorna-szám leválasztása
DDCE  A8        TAY
DDCF  B9 EC 00  LDA   00EC,Y
DDD2  29 01     AND   #01       egységszám leválasztása
DDD4  85 70     STA   70
DDD6  AE 53 02  LDX   0253
DDD9  B5 E2     LDA   E2,X
DDDB  29 01     AND   #01       egységszám leválasztása
DDDD  C5 70     CMP   70        azonos egység?
DDDF  D0 E1     BNE   DDC2      nem
DDE1  B9 60 02  LDA   0260,Y    tartalomjegyzék szektor-szám
DDE4  D5 D8     CMP   D8,X      azonos a file-éval?
DDE6  D0 DA     BNE   DDC2      nem
DDE8  B9 66 02  LDA   0266,Y
```

```
DDEB  D5 DD       CMP   DD,X       azonos a mutató?
DDED  D0 D3       BNE   DDC2       nem
DDEF  18          CLC
DDF0  60          RTS

*****************************      egy blokk felírása a relatív file-ba
DDF1  20 9E DF    JSR   DF9E       a puffer-szám betöltése
DDF4  50 06       BVC   DDFC       nem relatív file?
DDF6  20 5E DE    JSR   DE5E       blokk felírása
DDF9  20 99 D5    JSR   D599       és ellenőrzése
DDFC  60          RTS

*****************************      folytatósáv felírása
DDFD  20 2B DE    JSR   DE2B       a puffer mutató beállítása
DE00  A5 80       LDA   80         sáv-szám
DE02  91 94       STA   (94),Y     a pufferben
DE04  C8          INY
DE05  A5 81       LDA   81         szektor-szám
DE07  91 94       STA   (94),Y     a pufferben
DE09  4C 05 E1    JMP   E105       a relatív kapcsoló beállítása

*****************************      folytatósáv és szektor-szám beolvasása
DE0C  20 2B DE    JSR   DE2B       a puffer mutató beállítása
DE0F  B1 94       LDA   (94),Y     sáv-szám
DE11  85 80       STA   80
DE13  C8          INY
DE14  B1 94       LDA   (94),Y     és a szektor-szám olvasása
DE16  85 81       STA   81
DE18  60          RTS

*****************************      folytatósáv az utolsó blokknál
DE19  20 2B DE    JSR   DE2B       puffer mutató beállítása
DE1C  A9 00       LDA   #00        nullára
DE1E  91 94       STA   (94),Y     mint sáv-szám
DE20  C8          INY
DE21  A6 82       LDX   82         csatorna-szám
DE23  B5 C1       LDA   C1,X       mutató a blokkban
DE25  AA          TAX
DE26  CA          DEX              -1
DE27  8A          TXA
DE28  91 94       STA   (94),Y     mint mutató a blokkban
DE2A  60          RTS

*****************************      puffer mutató nullán
DE2B  20 93 DF    JSR   DF93       a puffer-szám beolvasása
DE2E  0A          ASL   A          2-szer
DE2F  AA          TAX
DE30  B5 9A       LDA   9A,X       puffer mutató /Hi/
DE32  85 95       STA   95
DE34  A9 00       LDA   #00
DE36  85 94       STA   94         puffer mutató /Lo/
DE38  A0 00       LDY   #00
DE3A  60          RTS

*****************************      a sáv és a szektor beolvasása
DE3B  20 EB D0    JSR   D0EB       csatorna-szám beolvasása
DE3E  20 93 DF    JSR   DF93       puffer-szám beolvasása
DE41  85 F9       STA   F9         és tárolása
DE43  0A          ASL   A          2-szer
DE44  A8          TAY
DE45  B9 06 00    LDA   0006,Y     sáv
DE48  85 80       STA   80
DE4A  B9 07 00    LDA   0007,Y     és a szektor olvasása a lemezről
DE4D  85 81       STA   81
DE4F  60          RTS
```

```
*****************************
DE50  A9 90      LDA  #90      írás-utasításkód
DE52  8D 4D 02   STA  024D
DE55  D0 28      BNE  DE7F
DE57  A9 80      LDA  #80      olvasás-utasításkód
DE59  8D 4D 02   STA  024D
DE5C  D0 21      BNE  DE7F
DE5E  A9 90      LDA  #90
DE60  8D 4D 02   STA  024D
DE63  D0 26      BNE  DE8B
DE65  A9 80      LDA  #80      olvasás-utasításkód
DE67  8D 4D 02   STA  024D
DE6A  D0 1F      BNE  DE8B
DE6C  A9 90      LDA  #90      írás-utasításkód
DE6E  8D 4D 02   STA  024D
DE71  D0 02      BNE  DE75
DE73  A9 80      LDA  #80      olvasás-utasításkód
DE75  8D 4D 02   STA  024D
DE78  A6 82      LDX  82       csatorna-szám
DE7A  B5 CD      LDA  CD,X     oldal-szektor puffer-szám
DE7C  AA         TAX
DE7D  10 13      BPL  DE92     hozzá van rendelve a puffer?
DE7F  20 D0 D6   JSR  D6D0     a fejléc generálása
DE82  20 93 DF   JSR  DF93     a puffer-szám beolvasása
DE85  AA         TAX
DE86  A5 7F      LDA  7F       egységszám
DE88  9D 5B 02   STA  025B,X
DE8B  20 15 E1   JSR  E115     puffer-szám
DE8E  20 93 DF   JSR  DF93     a pufferszám beolvasása
DE91  AA         TAX
DE92  4C 06 D5   JMP  D506     blokk felírása

*****************************  folytatósáv betöltése a pufferból
DE95  A9 00      LDA  #00
DE97  20 C8 D4   JSR  D4C8     puffer mutató nullán
DE9A  20 37 D1   JSR  D137     a byte beolvasása
DE9D  85 80      STA  80       és tárolása sávként
DE9F  20 37 D1   JSR  D137     a byte beolvasása
DEA2  85 81      STA  81       és tárolása szektorként
DEA4  60         RTS

*****************************  a puffer tartalmának lemásolása
DEA5  48         PHA
DEA6  A9 00      LDA  #00
DEA8  85 6F      STA  6F
DEAA  85 71      STA  71
DEAC  B9 E0 FE   LDA  FEE0,Y   Y puffercím /Hi/
DEAF  85 70      STA  70
DEB1  BD E0 FE   LDA  FEE0,X   X puffercím /Hi/
DEB4  85 72      STA  72
DEB6  68         PLA
DEB7  A8         TAY
DEB8  88         DEY
DEB9  B1 6F      LDA  (6F),Y   Y puffer tartalmát
DEBB  91 71      STA  (71),Y   X puffer szerint lemásolni!
DEBD  88         DEY
DEBE  10 F9      BPL  DEB9
DEC0  60         RTS
                               Y puffer törlése
DEC1  A8         TAY           puffer-szám
DEC2  B9 E0 FE   LDA  FEE0,Y   cím betöltése /Hi/
DEC5  85 70      STA  70
DEC7  A9 00      LDA  #00      cím betöltése /Lo/
DEC9  85 6F      STA  6F
DECB  A8         TAY
DECC  91 6F      STA  (6F),Y   puffer törlése
```

```
DECE  C8          INY
DECF  D0 FB       BNE   DECC
DED1  60          RTS

***************************   az oldal-szektor-számának betöltése
DED2  A9 00       LDA   #00
DED4  20 DC DE    JSR   DEDC       puffer mutató nullán
DED7  A0 02       LDY   #02
DED9  B1 94       LDA   (94),Y     a 2. byte tartalmazza az oldal-szektor számát
DEDB  60          RTS

***************************   puffer mutató beállítása az oldal-szektorra
DEDC  85 94       STA   94         mutató /Lo/
DEDE  A6 82       LDX   82         csatorna-szám
DEE0  B5 CD       LDA   CD,X       puffer-szám
DEE2  AA          TAX
DEE3  BD E0 FE    LDA   FEE0,X     puffer-cím /Hi/
DEE6  85 95       STA   95         beállítása
DEE8  60          RTS

***************************   oldal-szektor puffer mutatója
DEE9  48          PHA              mutató az oldal-szektorban
DEEA  20 DC DE    JSR   DEDC       a puffer mutató beállítása
DEED  48          PHA
DEEE  8A          TXA              puffer-szám
DEEF  0A          ASL   A          2-szer
DEF0  AA          TAX
DEF1  68          PLA              puffer mutató /Hi/
DEF2  95 9A       STA   9A,X
DEF4  68          PLA
DEF5  95 99       STA   99,X       puffer mutató /Lo/
DEF7  60          RTS

***************************   az oldal-szektor betöltése
DEF8  20 66 DF    JSR   DF66       a pufferben van az oldal-szektor?
DEFB  30 0E       BMI   DF0B       nincs
DEFD  50 13       BVC   DF12       ok
DEFF  A6 82       LDX   82         csatorna-szám
DF01  B5 CD       LDA   CD,X       puffer-szám
DF03  20 1B DF    JSR   DF1B       oldal-szektor olvasása
DF06  20 66 DF    JSR   DF66       a pufferben van?
DF09  10 07       BPL   DF12       igen?
DF0B  20 CB E1    JSR   E1CB       az utolsó oldal-szektor betöltése
DF0E  2C CE FE    BIT   FECE       a V-bit beállítása
DF11  60          RTS
DF12  A5 D6       LDA   D6         oldal-szektor végmutatója
DF14  20 E9 DE    JSR   DEE9       mutató beállítása az oldal-szektorban
DF17  2C CD FE    BIT   FECD       V-bit törlése
DF1A  60          RTS

***************************   oldal-szektor olvasása
DF1B  85 F9       STA   F9         puffer-szám
DF1D  A9 80       LDA   #80        olvasás-utasításkód
DF1F  D0 04       BNE   DF25

***************************   oldal-szektor felírása
DF21  85 F9       STA   F9         puffer-szám
DF23  A9 90       LDA   #90        írás-utasításkód
DF25  48          PHA
DF26  B5 EC       LDA   EC,X
DF28  29 01       AND   #01        egységszám leválasztása
DF2A  85 7F       STA   7F
DF2C  68          PLA
DF2D  05 7F       ORA   7F         utasításkód és az egységszám tárolása
DF2F  8D 4D 02    STA   024D
DF32  B1 94       LDA   (94),Y     sáv-szám
DF34  85 80       STA   80
```

```
DF36  C8        INY
DF37  B1 94     LDA  (94),Y      szektor-szám
DF39  85 81     STA  81
DF3B  A5 F9     LDA  F9          puffer-szám
DF3D  20 D3 D6  JSR  D6D3        paraméter átadása a vezérlőnek
DF40  A6 F9     LDX  F9
DF42  4C 93 D5  JMP  D593

*****************************    puffer mutató az oldal-szektorban
DF45  A6 82     LDX  82          csatorna-szám
DF47  B5 CD     LDA  CD,X        puffer-szám
DF49  4C EB D4  JMP  D4EB        puffer mutató beállítása

*****************************    egy relatív file blokkjainak száma
DF4C  A9 78     LDA  #78         oldal-szektoronként 12Ø blokkmutató!
DF4E  20 5C DF  JSR  DF5C        $7Ø/$71-hez
DF51  CA        DEX              oldal-szektor-szám
DF52  10 F8     BPL  DF4C        következő oldal-szektor?
DF54  A5 72     LDA  72          az utolsó blokkban lévő mutatók számát
DF56  4A        LSR  A           kettesével
DF57  20 5C DF  JSR  DF5C        hozzá kell adni az eddigi összeghez
DF5A  A5 73     LDA  73          a foglalt oldal-szektor blokkok száma
DF5C  18        CLC
DF5D  65 70     ADC  70
DF5F  85 70     STA  70          hozzáadás
DF61  90 02     BCC  DF65
DF63  E6 71     INC  71
DF65  60        RTS

*****************************    az oldal-szektor ellenőrzése a pufferben
DF66  20 D2 DE  JSR  DED2        az oldal-szektor számának betöltése
DF69  C5 D5     CMP  D5          egyenlő a szükséges blokk-számmal?
DF6B  D0 0E     BNE  DF7B        nem
DF6D  A4 D6     LDY  D6          mutató az oldal-szektorban
DF6F  B1 94     LDA  (94),Y      sáv-szám
DF71  F0 04     BEQ  DF77        még nincs behelyezve?
DF73  2C CD FE  BIT  FECD        bitek törlése
DF76  60        RTS

DF77  2C CF FE  BIT  FECF        N-bit beállítása
DF7A  60        RTS

DF7B  A5 D5     LDA  D5          oldal-szektor-szám nagyobb vagy
DF7D  C9 06     CMP  #06         egyenlő mint 6?
DF7F  B0 0A     BCS  DF8B        igen
DF81  0A        ASL  A
DF82  A8        TAY
DF83  A9 04     LDA  #04
DF85  85 94     STA  94
DF87  B1 94     LDA  (94),Y      sáv-szám beolvasása
DF89  D0 04     BNE  DF8F        már megtörtént?
DF8B  2C D0 FE  BIT  FED0        az N- és a V-bit beállítása
DF8E  60        RTS

DF8F  2C CE FE  BIT  FECE        a V-bit beállítása
DF92  60        RTS

*****************************    a puffer-szám betöltése
DF93  A6 82     LDX  82          csatorna-szám
DF95  B5 A7     LDA  A7,X        puffer-szám
DF97  10 02     BPL  DF9B        foglalt?
DF99  B5 AE     LDA  AE,X        puffer-szám a második táblázatból
DF9B  29 BF     AND  #BF         V-bit törlése
DF9D  60        RTS
```

```
DF9E  A6 82     LDX  82       csatorna-szám
DFA0  8E 57 02  STX  0257     tárolása
DFA3  B5 A7     LDA  A7,X     a puffer-szám betöltése
DFA5  10 09     BPL  DFB0     foglalt a puffer?
DFA7  8A        TXA
DFA8  18        CLC
DFA9  69 07     ADC  #07      a szám növelése 7-tel
DFAB  8D 57 02  STA  0257     és tárolása
DFAE  B5 AE     LDA  AE,X     puffer-szám a 2. táblázatból
DFB0  85 70     STA  70
DFB2  29 1F     AND  #1F      a felső 3. bit törlése
DFB4  24 70     BIT  70
DFB6  60        RTS

DFB7  A6 82     LDX  82       csatorna-szám
DFB9  B5 A7     LDA  A7,X     puffer-szám
DFBB  30 02     BMI  DFBF     szabad a puffer?
DFBD  B5 AE     LDA  AE,X     a 2. táblázatból származó puffer-szám
DFBF  C9 FF     CMP  #FF      szabad?
DFC1  60        RTS

DFC2  A6 82     LDX  82
DFC4  09 80     ORA  #80
DFC6  B4 A7     LDY  A7,X
DFC8  10 03     BPL  DFCD
DFCA  95 A7     STA  A7,X
DFCC  60        RTS

DFCD  95 AE     STA  AE,X
DFCF  60        RTS

****************************  a következő rekord átvitele a rel.file-ba
DFD0  A9 20     LDA  #20
DFD2  20 9D DD  JSR  DD9D     5. bit törlése
DFD5  A9 80     LDA  #80
DFD7  20 A6 DD  JSR  DDA6     7. bit tesztje
DFDA  D0 41     BNE  E01D     be van állítva?
DFDC  A6 82     LDX  82       csatorna-szám
DFDE  F6 B5     INC  B5,X     rekord-szám növelése
DFE0  D0 02     BNE  DFE4
DFE2  F6 BB     INC  BB,X     rekord-szám /Hi/
DFE4  A6 82     LDX  82       csatorna-szám
DFE6  B5 C1     LDA  C1,X     írásmutató
DFE8  F0 2E     BEQ  E018     nulla?
DFEA  20 E8 D4  JSR  D4E8     a puffer mutató beállítása
DFED  A6 82     LDX  82       csatorna-szám
DFEF  D5 C1     CMP  C1,X     kisebb a puffer mutató az irásmutatónál?
DFF1  90 03     BCC  DFF6     igen
DFF3  20 3C E0  JSR  E03C     blokk felírása, a következő blokk olvasása
DFF6  A6 82     LDX  82       csatorna-szám
DFF8  B5 C1     LDA  C1,X     írásmutató
DFFA  20 C8 D4  JSR  D4C8     a puffer mutató beállítása a fenti értékre
DFFD  A1 99     LDA  (99,X)   a byte a pufferból
DFFF  85 85     STA  85       az output-regiszterbe
E001  A9 20     LDA  #20
E003  20 9D DD  JSR  DD9D     az 5. bit törlése
E006  20 04 E3  JSR  E304     rekordhosszúság hozzáadása az írásmutatóhoz
E009  48        PHA           és tárolása
E00A  90 28     BCC  E034     a következő blokk?
E00C  A9 00     LDA  #00
E00E  20 F6 D4  JSR  D4F6     a sáv-szám betöltése
E011  D0 21     BNE  E034     megvan a blokk?
E013  68        PLA           mutató
E014  C9 02     CMP  #02      2-vel egyenlő
E016  F0 12     BEQ  E02A     igen
E018  A9 80     LDA  #80
E01A  20 97 DD  JSR  DD97     a 7. bit beállítása
```

```
E01D  20 2F D1   JSR  D12F     byte a pufferből
E020  B5 99      LDA  99,X     puffer mutató
E022  99 44 02   STA  0244,Y   mint vég-mutató
E025  A9 0D      LDA  #0D      CR
E027  85 85      STA  85       az output-regiszterben
E029  60         RTS

E02A  20 35 E0   JSR  E035
E02D  A6 82      LDX  82       csatorna-szám
E02F  A9 00      LDA  #00
E031  95 C1      STA  C1,X     írásmutató nullán
E033  60         RTS

E034  68         PLA
E035  A6 82      LDX  82       csatorna-szám
E037  95 C1      STA  C1,X     az írásmutató beállítása
E039  4C 6E E1   JMP  E16E

*****************************  blokk felírása és a következő blokk olvasása
E03C  20 D3 D1   JSR  D1D3     egységszám beolvasása
E03F  20 95 DE   JSR  DE95     a sáv- és szektor-szám beolvasása
E042  20 9E DF   JSR  DF9E     a puffer-szám beolvasása
E045  50 16      BVC  E05D     relatív file?
E047  20 5E DE   JSR  DE5E     blokk írása
E04A  20 1E CF   JSR  CF1E     puffer-váltás
E04D  A9 02      LDA  #02
E04F  20 C8 D4   JSR  D4C8     puffer mutató kettőn
E052  20 AB DD   JSR  DDAB     írás-utasításkód?
E055  D0 24      BNE  E07B     nem
E057  20 57 DE   JSR  DE57     blokk olvasása
E05A  4C 99 D5   JMP  D599     és ellenőrzése

E05D  20 1E CF   JSR  CF1E     puffer-váltás
E060  20 AB DD   JSR  DDAB     írás-utasításkód?
E063  D0 06      BNE  E06B     nem
E065  20 57 DE   JSR  DE57     blokk olvasása
E068  20 99 D5   JSR  D599     és ellenőrzése
E06B  20 95 DE   JSR  DE95     sáv- és szektor-szám beolvasása
E06E  A5 80      LDA  80       sáv
E070  F0 09      BEQ  E07B     folytató sáv?
E072  20 1E CF   JSR  CF1E     puffer-váltás
E075  20 57 DE   JSR  DE57     blokk olvasása
E078  20 1E CF   JSR  CF1E     puffer-váltás
E07B  60         RTS

*****************************  egy byte beírása a rekordba
E07C  20 05 E1   JSR  E105
E07F  20 93 DF   JSR  DF93     puffer-szám beolvasása
E082  0A         ASL  A        2-szer
E083  AA         TAX
E084  A5 85      LDA  85       adatbyte
E086  81 99      STA  (99,X)   beírása a pufferbe
E088  B4 99      LDY  99,X     puffer mutató
E08A  C8         INY           növelése
E08B  D0 09      BNE  E096     nem egyenlő nullával?
E08D  A4 82      LDY  82       csatorna-szám
E08F  B9 C1 00   LDA  00C1,Y   írásmutató
E092  F0 0A      BEQ  E09E     egyenlő nullával?
E094  A0 02      LDY  #02      puffer mutató 2-n
E096  98         TYA
E097  A4 82      LDY  82       csatorna-szám
E099  D9 C1 00   CMP  00C1,Y   puffer mutató azonos az írásmutatóval?
E09C  D0 05      BNE  E0A3     nem
E09E  A9 20      LDA  #20
E0A0  4C 97 DD   JMP  DD97     az 5. bit beállítása
```

```
E0A3  F6 99       INC   99,X     a puffer mutató növelése
E0A5  D0 03       BNE   E0AA     egyenlő nullával?
E0A7  20 3C E0    JSR   E03C     ha nem, blokk írása, a következő blokk olva-
E0AA  60          RTS            sása

*****************************    egy byte beírása a relativ file-ba
E0AB  A9 A0       LDA   #A0
E0AD  20 A6 DD    JSR   DDA6     a 6. és 7. bit tesztelése
E0B0  D0 27       BNE   E0D9     magas?
E0B2  A5 85       LDA   85       adatbyte
E0B4  20 7C E0    JSR   E07C     beírni a rekordba!
E0B7  A5 F8       LDA   F8       vége?
E0B9  F0 0D       BEQ   E0C8     igen
E0BB  60          RTS

E0BC  A9 20       LDA   #20
E0BE  20 A6 DD    JSR   DDA6     5. bit tesztelése
E0C1  F0 05       BEQ   E0C8     alacsony?
E0C3  A9 51       LDA   #51      51, "overflow in record"
E0C5  8D 6C 02    STA   026C     a hibakapcsoló beállítása
E0C8  20 F3 E0    JSR   E0F3     a rekord megmaradó részét nullákkal kitölteni!
E0CB  20 53 E1    JSR   E153
E0CE  AD 6C 02    LDA   026C     be van állítva a hibakapcsoló?
E0D1  F0 03       BEQ   E0D6     nincs
E0D3  4C C8 C1    JMP   C1C8     a hibaüzenet előkészítése

E0D6  4C BC E6    JMP   E6BC     hibamentes végrehajtás

E0D9  29 80       AND   #80      a 7. bit magas?
E0DB  D0 05       BNE   E0E2     igen
E0DD  A5 F8       LDA   F8
E0DF  F0 DB       BEQ   E0BC     vége?
E0E1  60          RTS

E0E2  A5 85       LDA   85       adatbyte
E0E4  48          PHA
E0E5  20 1C E3    JSR   E31C     az oldal-szektor bővítése
E0E8  68          PLA
E0E9  85 85       STA   85
E0EB  A9 80       LDA   #80
E0ED  20 9D DD    JSR   DD9D     a 7. bit törlése
E0F0  4C B2 E0    JMP   E0B2     byte beírása a file-ba

*****************************    rekord feltöltése nullákkal
E0F3  A9 20       LDA   #20
E0F5  20 A6 DD    JSR   DDA6     5. bit tesztelése
E0F8  D0 0A       BNE   E104     magas?
E0FA  A9 00       LDA   #00
E0FC  85 85       STA   85       adatbyteként nullát
E0FE  20 7C E0    JSR   E07C     beírni a rekordba,
E101  4C F3 E0    JMP   E0F3     amig a rekord megtelik!

E104  60          RTS

*****************************    puffer-szám beírása a táblázatba
E105  A9 40       LDA   #40
E107  20 97 DD    JSR   DD97     a 6. bit beállítása
E10A  20 9E DF    JSR   DF9E     a puffer-szám betöltése
E10D  09 40       ORA   #40      a 6. bit beállítása
E10F  AE 57 02    LDX   0257     + 7 csatorna-számot
E112  95 A7       STA   A7,X     beírni a táblázatba
E114  60          RTS

E115  20 9E DF    JSR   DF9E     puffer-szám olvasása
E118  29 BF       AND   #BF      6. bit törlése
E11A  AE 57 02    LDX   0257     csatorna-szám + 7
E11D  95 A7       STA   A7,X     beírása a táblázatba
E11F  60          RTS
```

```
****************************       egy byte beolvasása a relatív file-ból
E120  A9 80      LDA  #80
E122  20 A6 DD   JSR  DDA6      a 7. bit tesztelése
E125  D0 37      BNE  E15E      magas?
E127  20 2F D1   JSR  D12F      a byte beolvasása a pufferból
E12A  B5 99      LDA  99,X      puffer mutató
E12C  D9 44 02   CMP  0244,Y    összehasonlítása a vég-mutatóval
E12F  F0 22      BEQ  E153      azonos?
E131  F6 99      INC  99,X      puffer mutató növelése
E133  D0 06      BNE  E13B      nem egyenlő nullával?
E135  20 3C E0   JSR  E03C      blokk felírása, következő blokk olvasása
E138  20 2F D1   JSR  D12F      a byte olvasása a pufferból
E13B  A1 99      LDA  (99,X)
E13D  99 3E 02   STA  023E,Y    az output-regiszterbe
E140  A9 89      LDA  #89
E142  99 F2 00   STA  00F2,Y    a READ- és a WRITE-kapcsoló beállítása
E145  B5 99      LDA  99,X      puffer mutató
E147  D9 44 02   CMP  0244,Y    összehasonlítása a vég-mutatóval
E14A  F0 01      BEQ  E14D      azonos?
E14C  60         RTS

E14D  A9 81      LDA  #81
E14F  99 F2 00   STA  00F2,Y    vég-kapcsoló beállítása
E152  60         RTS

E153  20 D0 DF   JSR  DFD0      a következő rekord megkeresése
E156  20 2F D1   JSR  D12F      puffer-szám és csatorna-szám beolvasása
E159  A5 85      LDA  85        adatbyte
E15B  4C 3D E1   JMP  E13D      az output-regiszterbe

E15E  A6 82      LDX  82        csatorna-szám
E160  A9 0D      LDA  #0D       CR
E162  9D 3E 02   STA  023E,X    az output-regiszterbe
E165  A9 81      LDA  #81
E167  95 F2      STA  F2,X      a vég-kapcsoló beállítása
E169  A9 50      LDA  #50
E16B  20 C8 C1   JSR  C1C8      50, "record not present"
E16E  A6 82      LDX  82        csatorna-szám
E170  B5 C1      LDA  C1,X      írásmutató
E172  85 87      STA  87        tárolása
E174  C6 87      DEC  87
E176  C9 02      CMP  #02       egyenlő 2-vel?
E178  D0 04      BNE  E17E      nem
E17A  A9 FF      LDA  #FF
E17C  85 87      STA  87
E17E  B5 C7      LDA  C7,X      rekordhosszúság
E180  85 88      STA  88
E182  20 E8 D4   JSR  D4E8      a puffer mutató beállítása
E185  A6 82      LDX  82        csatorna-szám
E187  C5 87      CMP  87        puffer mutató nagyobb az írásmutatónál?
E189  90 19      BCC  E1A4
E18B  F0 17      BEQ  E1A4      nem
E18D  20 1E CF   JSR  CF1E      puffer-váltás
E190  20 B2 E1   JSR  E1B2
E193  90 08      BCC  E19D
E195  A6 82      LDX  82        csatorna-szám
E197  9D 44 02   STA  0244,X
E19A  4C 1E CF   JMP  CF1E      puffer-váltás

E19D  20 1E CF   JSR  CF1E      puffer-váltás
E1A0  A9 FF      LDA  #FF
E1A2  85 87      STA  87
E1A4  20 B2 E1   JSR  E1B2
E1A7  B0 03      BCS  E1AC
E1A9  20 E8 D4   JSR  D4E8      a puffer mutató beállítása
E1AC  A6 82      LDX  82        csatorna-szám
E1AE  9D 44 02   STA  0244,X    vég-mutató
E1B1  60         RTS
```

```
E1B2  20 2B DE   JSR  DE2B     puffer mutató nullán
E1B5  A4 87      LDY  87
E1B7  B1 94      LDA  (94),Y   a pufferból származó byte
E1B9  D0 0D      BNE  E1C8     nem egyenlő nullával?
E1BB  88         DEY
E1BC  C0 02      CPY  #02
E1BE  90 04      BCC  E1C4
E1C0  C6 88      DEC  88
E1C2  D0 F3      BNE  E1B7
E1C4  C6 88      DEC  88
E1C6  18         CLC
E1C7  60         RTS

E1C8  98         TYA
E1C9  38         SEC
E1CA  60         RTS

****************************   az utolsó oldal-szektor beolvasása
E1CB  20 D2 DE   JSR  DED2     az oldal-szektor számának beolvasása
E1CE  85 D5      STA  D5       és tárolása
E1D0  A9 04      LDA  #04
E1D2  85 94      STA  94
E1D4  A0 0A      LDY  #0A
E1D6  D0 04      BNE  E1DC
E1D8  88         DEY
E1D9  88         DEY
E1DA  30 26      BMI  E202
E1DC  B1 94      LDA  (94),Y   az előző blokk sáv-száma még
E1DE  F0 F8      BEQ  E1D8     nincs felírva?
E1E0  98         TYA
E1E1  4A         LSR  A        a szám: 2
E1E2  C5 D5      CMP  D5       egyenlő az aktuális blokk számával?
E1E4  F0 09      BEQ  E1EF     igen
E1E6  85 D5      STA  D5       ha nem, rekord-számként tárolni!
E1E8  A6 82      LDX  82       csatorna-szám
E1EA  B5 CD      LDA  CD,X     puffer-szám
E1EC  20 1B DF   JSR  DF1B     blokk olvasása
E1EF  A0 00      LDY  #00
E1F1  84 94      STY  94       puffer mutató
E1F3  B1 94      LDA  (94),Y   sáv-szám
E1F5  D0 0B      BNE  E202     követi tovább a blokkot?
E1F7  C8         INY
E1F8  B1 94      LDA  (94),Y   a szektor-szám egyenlő a vég-mutatóval?
E1FA  A8         TAY
E1FB  88         DEY
E1FC  84 D6      STY  D6       a vég-mutató tárolása
E1FE  98         TYA
E1FF  4C E9 DE   JMP  DEE9     a puffer mutató beállítása

E202  A9 67      LDA  #67
E204  20 45 E6   JSR  E645     67, "illegal track or sector"

****************************   P-utasítás, "Record"
E207  20 B3 C2   JSR  C2B3     sor ellenőrzése
E20A  AD 01 02   LDA  0201     másodlagos cím
E20D  85 83      STA  83
E20F  20 EB D0   JSR  D0EB     a csatorna-szám keresése
E212  90 05      BCC  E219     megtalálta?
E214  A9 70      LDA  #70
E216  20 C8 C1   JSR  C1C8     70, "no block"
E219  A9 A0      LDA  #A0
E21B  20 9D DD   JSR  DD9D     6. és 7. bit törlése
E21E  20 25 D1   JSR  D125     a "REL"-file ellenőrzése
E221  F0 05      BEQ  E228     igen
E223  A9 64      LDA  #64
E225  20 C8 C1   JSR  C1C8     64, "file type mismatch"
E228  B5 EC      LDA  EC,X
```

```
E22A  29 01     AND  #01
E22C  85 7F     STA  7F        egységszám
E22E  AD 02 02  LDA  0202      rekord-szám /Lo/
E231  95 B5     STA  B5,X
E233  AD 03 02  LDA  0203      rekord-szám /Hi/
E236  95 BB     STA  BB,X
E238  A6 82     LDX  82        csatorna-szám
E23A  A9 89     LDA  #89
E23C  95 F2     STA  F2,X      READ- és WRITE-kapcsoló
E23E  AD 04 02  LDA  0204      a byte-mutató
E241  F0 10     BEQ  E253      nulla?
E243  38        SEC
E244  E9 01     SBC  #01
E246  F0 0B     BEQ  E253
E248  D5 C7     CMP  C7,X      összehasonlítani a file rekordhosszúságával!
E24A  90 07     BCC  E253
E24C  A9 51     LDA  #51
E24E  8D 6C 02  STA  026C      51, "overflow in record"
E251  A9 00     LDA  #00
E253  85 D4     STA  D4
E255  20 0E CE  JSR  CE0E      a rel.file-on belüli mutató kiszámítása
E258  20 F8 DE  JSR  DEF8      a megfelelő oldal-szektor-blokk olvasása
E25B  50 08     BVC  E265      megvan a blokk?
E25D  A9 80     LDA  #80
E25F  20 97 DD  JSR  DD97      a 7. bit beállítása
E262  4C 5E E1  JMP  E15E      és 50, "record not present"

E265  20 75 E2  JSR  E275
E268  A9 80     LDA  #80
E26A  20 A6 DD  JSR  DDA6      a 7. bit tesztelése
E26D  F0 03     BEQ  E272      alacsony
E26F  4C 5E E1  JMP  E15E      50, "record not present"

E272  4C 94 C1  JMP  C194      kész

E275  20 9C E2  JSR  E29C
E278  A5 D7     LDA  D7        mutató a relatív file-ban
E27A  20 C8 D4  JSR  D4C8      a puffer mutató beállítása
E27D  A6 82     LDX  82        csatorna-szám
E27F  B5 C7     LDA  C7,X      rekordhosszúság
E281  38        SEC
E282  E5 D4     SBC  D4        a pozició negatív
E284  B0 03     BCS  E289      pozitiv?
E286  4C 02 E2  JMP  E202      67, "illegal track or sector"

E289  18        CLC
E28A  65 D7     ADC  D7        mutató hozzáadása az adatblokkban
E28C  90 03     BCC  E291      nincs tulcsordulás
E28E  69 01     ADC  #01       plusz 2
E290  38        SEC
E291  20 09 E0  JSR  E009      mutató beállítása
E294  4C 38 E1  JMP  E138      és a byte beolvasása a pufferből

E297  A9 51     LDA  #51
E299  20 C8 C1  JSR  C1C8      51, "overflow in record"
E29C  A5 94     LDA  94        puffer mutató /Lo/
E29E  85 89     STA  89
E2A0  A5 95     LDA  95        puffer mutató /Hi/
E2A2  85 8A     STA  8A
E2A4  20 D0 E2  JSR  E2D0      sáv és szektor összehasonlítása
E2A7  D0 01     BNE  E2AA      nem egyenlő?
E2A9  60        RTS

E2AA  20 F1 DD  JSR  DDF1
E2AD  20 0C DE  JSR  DE0C
E2B0  A5 80     LDA  80        sáv
E2B2  F0 0E     BEQ  E2C2      folytatóblokk?
```

```
E2B4   20 D3 E2   JSR   E2D3      sáv és szektor-szám összehasonlítása
E2B7   D0 06      BNE   E2BF      nem egyenlő
E2B9   20 1E CF   JSR   CF1E      puffer-váltás
E2BC   4C DA D2   JMP   D2DA

E2BF   20 DA D2   JSR   D2DA
E2C2   A0 00      LDY   #00
E2C4   B1 89      LDA   (89),Y    sáv
E2C6   85 80      STA   80
E2C8   C8         INY
E2C9   B1 89      LDA   (89),Y    és a következő blokk szektora
E2CB   85 81      STA   81
E2CD   4C AF D0   JMP   D0AF      blokk olvasása

E2D0   20 3E DE   JSR   DE3E
E2D3   A0 00      LDY   #00
E2D5   B1 89      LDA   (89),Y    sáv-szám
E2D7   C5 80      CMP   80        összehasonlítása
E2D9   F0 01      BEQ   E2DC
E2DB   60         RTS

E2DC   C8         INY
E2DD   B1 89      LDA   (89),Y    szektor-szám
E2DF   C5 81      CMP   81        összehasonlítása
E2E1   60         RTS

******************************    adatblokk felosztása a rekordban
E2E2   20 2B DE   JSR   DE2B      puffer mutató beállítása
E2E5   A0 02      LDY   #02
E2E7   A9 00      LDA   #00
E2E9   91 94      STA   (94),Y    puffer törlése
E2EB   C8         INY
E2EC   D0 FB      BNE   E2E9
E2EE   20 04 E3   JSR   E304      mutató a következő rekordra!
E2F1   95 C1      STA   C1,X
E2F3   A8         TAY
E2F4   A9 FF      LDA   #FF
E2F6   91 94      STA   (94),Y    $FF, mint a rekord első karaktere
E2F8   20 04 E3   JSR   E304      a mutató a következő rekordra!
E2FB   90 F4      BCC   E2F1
E2FD   D0 04      BNE   E303      megtelt a blokk?
E2FF   A9 00      LDA   #00
E301   95 C1      STA   C1,X      írásmutató nullán
E303   60         RTS

******************************    mutató beállítása a következő rekordra
E304   A6 82      LDX   82        csatorna-szám
E306   B5 C1      LDA   C1,X      írásmutató
E308   38         SEC
E309   F0 0D      BEQ   E318      egyenlő nullával?
E30B   18         CLC
E30C   75 C7      ADC   C7,X      rekordhosszúság hozzáadása
E30E   90 0B      BCC   E31B      kisebb mint 256?
E310   D0 06      BNE   E318      egyenlő 256-tal?
E312   A9 02      LDA   #02
E314   2C CC FE   BIT   FECC
E317   60         RTS

E318   69 01      ADC   #01       kettőt hozzáadni
E31A   38         SEC
E31B   60         RTS

******************************    az oldal-szektor bővítése
E31C   20 D3 D1   JSR   D1D3      az egységszám betöltése
E31F   20 CB E1   JSR   E1CB      az utolsó oldal-szektor betöltése
E322   20 9C E2   JSR   E29C
E325   20 7B CF   JSR   CF7B
```

```
E328  A5 D6     LDA  D6
E32A  85 87     STA  87
E32C  A5 D5     LDA  D5         oldal-szektor száma
E32E  85 86     STA  86
E330  A9 00     LDA  #00
E332  85 88     STA  88
E334  A9 00     LDA  #00
E336  85 D4     STA  D4
E338  20 0E CE  JSR  CE0E       oldal-szektor-szám és a mutató kiszámitása!
E33B  20 4D EF  JSR  EF4D       szabad blokkok száma
E33E  A4 82     LDY  82         csatorna-szám
E340  B6 C7     LDX  C7,Y       rekordhosszúság
E342  CA        DEX
E343  8A        TXA
E344  18        CLC
E345  65 D7     ADC  D7         plusz az adatblokkban lévő mutató
E347  90 0C     BCC  E355
E349  E6 D6     INC  D6
E34B  E6 D6     INC  D6         mutatót a végén 2-vel növelni! /sáv/szektor/
E34D  D0 06     BNE  E355       nincs átvitel?
E34F  E6 D5     INC  D5         az oldal-szektor-számának növelése
E351  A9 10     LDA  #10
E353  85 D6     STA  D6         mutatót 16-ra állítani!
E355  A5 87     LDA  87
E357  18        CLC
E358  69 02     ADC  #02
E35A  20 E9 DE  JSR  DEE9       oldal-szektor puffer mutatójának beállítása
E35D  A5 D5     LDA  D5         oldal-szektor száma
E35F  C9 06     CMP  #06
E361  90 05     BCC  E368       kisebb mint 6?
E363  A9 52     LDA  #52
E365  20 C8 C1  JSR  C1C8       52, "file too large"
E368  A5 D6     LDA  D6         vég-mutató
E36A  38        SEC
E36B  E5 87     SBC  87         minusz az utolsó vég-mutató
E36D  B0 03     BCS  E372
E36F  E9 0F     SBC  #0F        minusz 16
E371  18        CLC
E372  85 72     STA  72
E374  A5 D5     LDA  D5         oldal-szektor száma
E376  E5 86     SBC  86         minusz az utolsó oldal-szektor száma
E378  85 73     STA  73         tárolás
E37A  A2 00     LDX  #00
E37C  86 70     STX  70         az eredmény törlése
E37E  86 71     STX  71
E380  AA        TAX
E381  20 51 DF  JSR  DF51       a relatív file-ok blokkszámának kiszámítása
E384  A5 71     LDA  71
E386  D0 07     BNE  E38F
E388  A6 70     LDX  70
E38A  CA        DEX
E38B  D0 02     BNE  E38F
E38D  E6 88     INC  88
E38F  CD 73 02  CMP  0273       relatív file blokkszáma
E392  90 09     BCC  E39D       nagyobb mint a lemezen levő szabad blokkok-?
E394  D0 CD     BNE  E363       52, "file too large"
E396  AD 72 02  LDA  0272
E399  C5 70     CMP  70
E39B  90 C6     BCC  E363       52, "file too large"
E39D  A9 01     LDA  #01
E39F  20 F6 D4  JSR  D4F6       byte a pufferból
E3A2  18        CLC
E3A3  69 01     ADC  #01        plusz 1
E3A5  A6 82     LDX  82
E3A7  95 C1     STA  C1,X       mint írásmutató
E3A9  20 1E F1  JSR  F11E       szabad blokk keresése a BAM-ban
E3AC  20 FD DD  JSR  DDFD       sáv és szektor a pufferben
```

```
E3AF   A5 88      LDA   88
E3B1   D0 15      BNE   E3C8      csak egy blokk szükséges?
E3B3   20 5E DE   JSR   DE5E      blokk beírása
E3B6   20 1E CF   JSR   CF1E      puffer-váltás
E3B9   20 D0 D6   JSR   D6D0      paraméter átadása a lemezvezérlőnek
E3BC   20 1E F1   JSR   F11E      szabad blokk keresése a BAM-ban
E3BF   20 FD DD   JSR   DDFD      sáv és szektor a pufferben
E3C2   20 E2 E2   JSR   E2E2      puffer törlése
E3C5   4C D4 E3   JMP   E3D4

E3C8   20 1E CF   JSR   CF1E      puffer-váltás
E3CB   20 D0 D6   JSR   D6D0      paraméter átadása a lemezvezérlőnek
E3CE   20 E2 E2   JSR   E2E2      puffer törlése
E3D1   20 19 DE   JSR   DE19      nulla byte és a végmutató a pufferben
E3D4   20 5E DE   JSR   DE5E      blokk felírása
E3D7   20 0C DE   JSR   DE0C      a sáv és a szektor betöltése
E3DA   A5 80      LDA   80        sáv
E3DC   48         PHA
E3DD   A5 81      LDA   81        és szektor
E3DF   48         PHA             tárolása
E3E0   20 3E DE   JSR   DE3E      sáv és szektor betöltése
E3E3   A5 81      LDA   81        a szektor
E3E5   48         PHA
E3E6   A5 80      LDA   80        és a sáv tárolása
E3E8   48         PHA
E3E9   20 45 DF   JSR   DF45      oldal-szektor puffer mutatójának beállítása
E3EC   AA         TAX
E3ED   D0 0A      BNE   E3F9      mutató nem egyenlő nullával?
E3EF   20 4E E4   JSR   E44E      oldal-szektor felírása, következő beolvasása
E3F2   A9 10      LDA   #10
E3F4   20 E9 DE   JSR   DEE9      puffer mutató 16-on
E3F7   E6 86      INC   86        oldal-szektor számának növelése
E3F9   68         PLA
E3FA   20 8D DD   JSR   DD8D      sáv az oldal-szektorban
E3FD   68         PLA
E3FE   20 8D DD   JSR   DD8D      szektor az oldal-szektorban
E401   68         PLA
E402   85 81      STA   81        szektor
E404   68         PLA
E405   85 80      STA   80        és a sáv visszaolvasása
E407   F0 0F      BEQ   E418      nincs további blokk
E409   A5 86      LDA   86        oldal-szektor száma
E40B   C5 D5      CMP   D5        megváltozott?
E40D   D0 A7      BNE   E3B6      igen
E40F   20 45 DF   JSR   DF45      puffer mutató felírása az oldal-szektorba
E412   C5 D6      CMP   D6        végmutató
E414   90 A0      BCC   E3B6      kisebb?
E416   F0 B0      BEQ   E3C8      azonos
E418   20 45 DF   JSR   DF45      puffer mutató beírása az oldal-szektorba
E41B   48         PHA
E41C   A9 00      LDA   #00
E41E   20 DC DE   JSR   DEDC      puffer mutató nullán
E421   A9 00      LDA   #00
E423   A8         TAY
E424   91 94      STA   (94),Y    nulla, mint sáv-szám
E426   C8         INY
E427   68         PLA             vég-mutató
E428   38         SEC
E429   E9 01      SBC   #01       minusz egy
E42B   91 94      STA   (94),Y    mint szektor
E42D   20 6C DE   JSR   DE6C      blokk felírása
E430   20 99 D5   JSR   D599      és ellenőrzése
E433   20 F4 EE   JSR   EEF4      a BAM aktualizálása
E436   20 0E CE   JSR   CE0E      relativ file mutatójának aktualizálása
E439   20 1E CF   JSR   CF1E      puffer-váltás
```

```
E43C  20 F8 DE   JSR  DEF8      helyes oldal-szektor?
E43F  70 03      BVS  E444      nem
E441  4C 75 E2   JMP  E275

E444  A9 80      LDA  #80
E446  20 97 DD   JSR  DD97      a 7. bit beállítása
E449  A9 50      LDA  #50
E44B  20 C8 C1   JSR  C1C8      5Ø, "record not present"

****************************    oldal-szektor felírása és új olvasása
E44E  20 1E F1   JSR  F11E      szabad blokk a BAM-ban
E451  20 1E CF   JSR  CF1E      puffer-váltás
E454  20 F1 DD   JSR  DDF1      blokk írása
E457  20 93 DF   JSR  DF93      puffer-szám beolvasása
E45A  48         PHA
E45B  20 C1 DE   JSR  DEC1      puffer törlése
E45E  A6 82      LDX  82        csatorna-szám
E460  B5 CD      LDA  CD,X      puffer-szám
E462  A8         TAY
E463  68         PLA
E464  AA         TAX
E465  A9 10      LDA  #10       oldal-szektor 16 byte átmásolása
E467  20 A5 DE   JSR  DEA5      a pufferba
E46A  A9 00      LDA  #00
E46C  20 DC DE   JSR  DEDC      puffer mutató nullán, korábbi oldal-
E46F  A0 02      LDY  #02
E471  B1 94      LDA  (94),Y    oldal-szektor száma
E473  48         PHA
E474  A9 00      LDA  #00
E476  20 C8 D4   JSR  D4C8      puffer mutató nullán, új oldal-szektor
E479  68         PLA
E47A  18         CLC
E47B  69 01      ADC  #01       oldal-szektor számának növelése
E47D  91 94      STA  (94),Y
E47F  0A         ASL  A         2-szer
E480  69 04      ADC  #04       plusz 4
E482  85 89      STA  89        egyenlő a sáv/szektor-on álló mutatóval
E484  A8         TAY
E485  38         SEC
E486  E9 02      SBC  #02       minusz 2
E488  85 8A      STA  8A        egyenlő az előző oldal-szektor mutatóval?
E48A  A5 80      LDA  80        sáv
E48C  85 87      STA  87
E48E  91 94      STA  (94),Y    a pufferben
E490  C8         INY
E491  A5 81      LDA  81        szektor
E493  85 88      STA  88
E495  91 94      STA  (94),Y    a pufferben
E497  A0 00      LDY  #00
E499  98         TYA
E49A  91 94      STA  (94),Y    nulla a pufferben
E49C  C8         INY
E49D  A9 11      LDA  #11       17
E49F  91 94      STA  (94),Y    byte-ok száma a blokkban
E4A1  A9 10      LDA  #10       16
E4A3  20 C8 D4   JSR  D4C8      puffer mutató 16-on
E4A6  20 50 DE   JSR  DE50      blokk felírása
E4A9  20 99 D5   JSR  D599      és ellenőrzése
E4AC  A6 82      LDX  82        csatorna-szám
E4AE  B5 CD      LDA  CD,X      oldal-szektor puffer-száma
E4B0  48         PHA
E4B1  20 9E DF   JSR  DF9E      puffer-szám beolvasása
E4B4  A6 82      LDX  82        csatorna-szám beírása
E4B6  95 CD      STA  CD,X      a táblázatba
E4B8  68         PLA
E4B9  AE 57 02   LDX  0257      csatorna-szám + 7
E4BC  95 A7      STA  A7,X      a táblázatba
```

```
E4BE   A9 00          LDA   #00
E4C0   20 C8 D4       JSR   D4C8        puffer mutató nullán
E4C3   A0 00          LDY   #00
E4C5   A5 80          LDA   80          sáv
E4C7   91 94          STA   (94),Y      a pufferben
E4C9   C8             INY
E4CA   A5 81          LDA   81          szektor
E4CC   91 94          STA   (94),Y      a pufferben
E4CE   4C DE E4       JMP   E4DE

E4D1   20 93 DF       JSR   DF93        a puffer-szám beolvasása
E4D4   A6 82          LDX   82          csatorna-szám
E4D6   20 1B DF       JSR   DF1B        blokk olvasása
E4D9   A9 00          LDA   #00
E4DB   20 C8 D4       JSR   D4C8        puffer mutató nullán
E4DE   C6 8A          DEC   8A
E4E0   C6 8A          DEC   8A          oldal-szektor blokkok számlálója
E4E2   A4 89          LDY   89
E4E4   A5 87          LDA   87          sáv-szám
E4E6   91 94          STA   (94),Y      a pufferben
E4E8   C8             INY
E4E9   A5 88          LDA   88          szektor-szám
E4EB   91 94          STA   (94),Y      a pufferben
E4ED   20 5E DE       JSR   DE5E        blokk felírása
E4F0   20 99 D5       JSR   D599        és ellenőrzése
E4F3   A4 8A          LDY   8A          oldal-szektor blokkok számlálója
E4F5   C0 03          CPY   #03
E4F7   B0 D8          BCS   E4D1        nagyobb vagy egyenlő mint 3?
E4F9   4C 1E CF       JMP   CF1E        puffer-váltás

****************************            hibaüzenetek táblázata
E4FC   00                               ØØ
E4FD   A0 4F CB                         "oK"
E500   20 21 22 23 24 27                "read error" hiba száma
E506   D2 45 41 44                      "Read"
E50A   89                               mutató "error"-ra
E50B   52                               52
E50C   83                               mutató"file"-ra
E50D   20 54 4F 4F 20 4C 41 52          "too largE"
E515   47 C5
E517   50                               5Ø
E518   8B 06                            mutató "record"-on és "not"
E51A   20 50 52 45 53 45 4E D4          "presenT"-en
E522   51                               51
E523   CF 56 45 52 46 4C 4F 57          "overflow in"
E52B   20 49 4E
E52E   8B                               mutató a "rekord"-on
E52F   25 28                            "write error" hiba száma
E531   8A 89                            mutató a "write"-on és "error"-on
E533   26                               26
E534   8A                               mutató a "write"-on
E535   20 50 52 4F 54 45 43 54          "protect oN"
E53D   20 4F CE
E540   29 88                            29
E542   20 49 44                         mutató "disk id"-n
E545   85                               mutató "mismatch"-n
E546   30 31 32 33 34                   "syntax error" hiba számai
E54B   D3 59 4E 54 41                   "Syntax"
E550   58
E551   89                               mutató "error"-on
E552   60                               6Ø
E553   8A 03 84                         mutató "write"-n "file"-n és "open"-en
E556   63 83                            63 mutató "file"-on
E558   20 45 58 49 53 54 D3             "existS"-en
E55F   64                               64
E560   83                               mutató "file"-
E561   20 54 59 50 45                   type"-on
```

```
E566  85                               mutató "mismatch"-on
E567  65 CE                            65
E569  4F 20 42 4C 4F 43 CB 66          "no block"
E571  67                               "illegal track or sector" hiba számai
E572  C9 4C 4C 45 47 41 4C 20          "illegal"
E57A  54 52 41 43 4B 20 4F 52          "track or"
E582  20 53 45 43 54 4F D2             "sectoR"
E589  61                               61
E58A  83 06 84                         mutató a "file"-on, "not"-on és "open"-en
E58D  39 62 83                         "file not found" hiba száma
E590  06 87 01                         mutató "file"-on, "not"-on és "found"-on
E593  83                               mutató "file"-on
E594  53                               "s
E595  20 53 43 52 41 54 43 48           scratch
E59D  45 C4                            eD"
E59F  70                               7Ø
E5A0  CE 4F 20 43 48 41 4E 4E          "no channeL"
E5A8  45 CC
E5AA  71                               71
E5AB  C4 49 52                         "Dir"
E5AE  89                               mutató "error"-on
E5AF  72                               72
E5B0  88                               mutató "disk"-en
E5B1  20 46 55 4C CC                   " fulL"
E5B6  73                               73
E5B7  C3 42 4D 20 44 4F 53 20          "Cbm dos"
E5BF  56 32 2E 36 20                   " v2.6
E5C4  31
E5C5  35 34 B1 74 C4                   1541
E5CA  52
E5CB  49 56 45 06 20 52                "drive" "not"
E5D1  45 41 44 D9 09                   " readY"
E5D6  C5 52 52 4F D2                   "ErroR"
E5DB  0A
E5DC  D7 52 49 54 C5                   "WritE"
E5E1  03
E5E2  C6 49 4C C5                      "FilE"
E5E6  04
E5E7  CF 50 45 CE                      "OpeN"
E5EB  05 CD
E5ED  49 53 4D 41 54 43 C8 06          "MismatcH"
E5F5  CE 4F D4                         "NoT"
E5F8  07
E5F9  C6 4F 55 4E C4                   "FounD"
E5FE  08
E5FF  C4 49 53 CB                      "DisK"
E603  0B
E604  D2 45 43 4F 52 C4                "RecorD"

****************************           a hiba-szám és a hibaüzenet előkészítése
E60A  48          PHA                  a hibakód tárolása
E60B  86 F9       STX   F9             egységszám
E60D  8A          TXA
E60E  0A          ASL   A              2-szer
E60F  AA          TAX                  mutató
E610  B5 06       LDA   06,X
E612  85 80       STA   80             a sáv
E614  B5 07       LDA   07,X
E616  85 81       STA   81             és a szektor-szám beolvasása
E618  68          PLA                  a hibakód visszaolvasása
E619  29 0F       AND   #0F            Ø-3 bit leválasztása
E61B  F0 08       BEQ   E625           ha nulla, akkor 24, "read error"
E61D  C9 0F       CMP   #0F            15?
E61F  D0 06       BNE   E627
E621  A9 74       LDA   #74
E623  D0 08       BNE   E62D           74, "drive not ready"
E625  A9 06       LDA   #06            6
```

```
E627  09 20     ORA  #20       $20 hozzáadása
E629  AA        TAX
E62A  CA        DEX
E62B  CA        DEX            -2
E62C  8A        TXA
E62D  48        PHA            a hiba-szám tárolása
E62E  AD 2A 02  LDA  022A      lemezművelet száma
E631  C9 00     CMP  #00       OPEN vagy VALIDATE?
E633  D0 0F     BNE  E644      nem
E635  A9 FF     LDA  #FF
E637  8D 2A 02  STA  022A
E63A  68        PLA            a hiba-szám visszaolvasása
E63B  20 C7 E6  JSR  E6C7      a hibaüzenet generálása
E63E  20 42 D0  JSR  D042      a BAM betöltése
E641  4C 48 E6  JMP  E648      a hibaüzenet előkészítése

E644  68        PLA
E645  20 C7 E6  JSR  E6C7
E648  20 BD C1  JSR  C1BD      input-puffer törlése
E64B  A9 00     LDA  #00
E64D  8D F9 02  STA  02F9      a hiba-kapcsoló törlése
E650  20 2C C1  JSR  C12C      LED kikapcsolása
E653  20 DA D4  JSR  D4DA      a 17. és 18. csatorna lezárása
E656  A9 00     LDA  #00
E658  85 A3     STA  A3        input-puffer mutatója nullán
E65A  A2 45     LDX  #45
E65C  9A        TXS            a verem-mutató inicializálása
E65D  A5 84     LDA  84        másodlagos cím
E65F  29 0F     AND  #0F
E661  85 83     STA  83
E663  C9 0F     CMP  #0F       15 ?
E665  F0 31     BEQ  E698      igen, csatornautasítása
E667  78        SEI
E668  A5 79     LDA  79        aktiv a LISTEN?
E66A  D0 1C     BNE  E688      igen
E66C  A5 7A     LDA  7A        aktiv a TALK?
E66E  D0 10     BNE  E680      igen
E670  A6 83     LDX  83        csatorna-szám
E672  BD 2B 02  LDA  022B,X    nyiva a csatorna ehhez a másodlagos címhez?
E675  C9 FF     CMP  #FF
E677  F0 1F     BEQ  E698      nincs
E679  29 0F     AND  #0F
E67B  85 82     STA  82        csatorna-szám
E67D  4C 8E E6  JMP  E68E

*****************************  TALK
E680  20 EB D0  JSR  D0EB      olvasás-csatorna megnyitása
E683  EA        NOP            egy byte fogadása
E684  EA        NOP
E685  EA        NOP
E686  D0 06     BNE  E68E

*****************************  LISTEN
E688  20 07 D1  JSR  D107      írás-csatorna megnyitása
E68B  EA        NOP            egy byte fogadása
E68C  EA        NOP
E68D  EA        NOP
E68E  20 25 D1  JSR  D125      file-tipus ellenőrzése
E691  C9 04     CMP  #04       relativ file-tipus?
E693  B0 03     BCS  E698      igen
E695  20 27 D2  JSR  D227      csatorna lezárása
E698  4C E7 EB  JMP  EBE7

*****************************  hexadecimális szám átalakítása decimálissá
E69B  AA        TAX
E69C  A9 00     LDA  #00
E69E  F8        SED
```

```
E69F   E0 00      CPX  #00
E6A1   F0 07      BEQ   E6AA     hexadecimális szám átalakítása BCD-számmá
E6A3   18         CLC
E6A4   69 01      ADC  #01
E6A6   CA         DEX
E6A7   4C 9F E6   JMP   E69F

E6AA   D8         CLD

******************************   BCD-szám felbontása két byte-ra
E6AB   AA         TAX
E6AC   4A         LSR   A
E6AD   4A         LSR   A        hi-Nibble-t eltolása lefelé
E6AE   4A         LSR   A
E6AF   4A         LSR   A
E6B0   20 B4 E6   JSR   E6B4     átalakítás ASCII
E6B3   8A         TXA
E6B4   29 0F      AND  #0F       legfelső 4 bit törlése
E6B6   09 30      ORA  #30       "Ø" hozzáírása
E6B8   91 A5      STA  (A5),Y    a pufferbe beírni!
E6BA   C8         INY            puffer-mutató növelése
E6BB   60         RTS

******************************   "ok" beírása a pufferbe
E6BC   20 23 C1   JSR   C123     hiba-kapcsolók törlése
E6BF   A9 00      LDA  #00       Ø hiba-szám
E6C1   A0 00      LDY  #00
E6C3   84 80      STY   80       Ø sáv
E6C5   84 81      STY   81       Ø szektor

******************************   hibaüzenet a pufferben /szám az akku-ban/
E6C7   A0 00      LDY  #00       puffer-mutató
E6C9   A2 D5      LDX  #D5
E6CB   86 A5      STX   A5       $A5/$A6 mutató $2D5-n
E6CD   A2 02      LDX  #02
E6CF   86 A6      STX   A6
E6D1   20 AB E6   JSR   E6AB     ASCII hiba-szám és a
E6D4   A9 2C      LDA  #2C       ","
E6D6   91 A5      STA  (A5),Y    beírása a pufferbe
E6D8   C8         INY            puffer-mutató növelése
E6D9   AD D5 02   LDA   02D5     lemezstatus első számjegye
E6DC   8D 43 02   STA   0243     az output-regiszterbe!
E6DF   8A         TXA            hiba-szám az akku-ban
E6E0   20 06 E7   JSR   E706     hibaüzenet a pufferben
E6E3   A9 2C      LDA  #2C       ","
E6E5   91 A5      STA  (A5),Y    beírása a pufferbe
E6E7   C8         INY            a puffer-mutató növelése
E6E8   A5 80      LDA   80       sáv
E6EA   20 9B E6   JSR   E69B
E6ED   A9 2C      LDA  #2C
E6EF   91 A5      STA  (A5),Y
E6F1   C8         INY
E6F2   A5 81      LDA   81       szektor
E6F4   20 9B E6   JSR   E69B
E6F7   88         DEY
E6F8   98         TYA
E6F9   18         CLC
E6FA   69 D5      ADC  #D5
E6FC   8D 49 02   STA   0249     vég-mutató
E6FF   E6 A5      INC   A5
E701   A9 88      LDA  #88       a READ-kapcsoló beállítása
E703   85 F7      STA   F7
E705   60         RTS

******************************   hibaüzenet szövegének beírása a pufferbe
E706   AA         TAX            X szerinti hibakód
E707   A5 86      LDA   86
```

```
E709  48        PHA             a $86/$87 mutató tárolása
E70A  A5 87     LDA   87
E70C  48        PHA
E70D  A9 FC     LDA   #FC
E70F  85 86     STA   86        $86/$87 mutató $E4FC-n
E711  A9 E4     LDA   #E4       hibaüzenetek kezdete
E713  85 87     STA   87
E715  8A        TXA             hiba-szám az akku-ban
E716  A2 00     LDX   #00
E718  C1 86     CMP   (86,X)    összehasonlítás a hiba-számmal a táblázatban
E71A  F0 21     BEQ   E73D
E71C  48        PHA
E71D  20 75 E7  JSR   E775      7. bit a Carry-be
E720  90 05     BCC   E727      nincs beállítva?
E722  20 75 E7  JSR   E775      7. bit a Carry-be
E725  90 FB     BCC   E722      várakozás a karakterre
E727  A5 87     LDA   87
E729  C9 E6     CMP   #E6
E72B  90 08     BCC   E735      $E60A táblázat vége?
E72D  D0 0A     BNE   E739
E72F  A9 0A     LDA   #0A
E731  C5 86     CMP   86
E733  90 04     BCC   E739
E735  68        PLA
E736  4C 18 E7  JMP   E718      nem, folytatni

E739  68        PLA
E73A  4C 4D E7  JMP   E74D      kész

E73D  20 67 E7  JSR   E767      egy karakter beolvasása, 7. bit a Carry-be
E740  90 FB     BCC   E73D      várakozás a karakterre
E742  20 54 E7  JSR   E754      és felírása a pufferbe
E745  20 67 E7  JSR   E767      következő karakter beolvasása
E748  90 F8     BCC   E742      várakozás a karakterre
E74A  20 54 E7  JSR   E754      karakter a pufferben
E74D  68        PLA
E74E  85 87     STA   87
E750  68        PLA             a $86/$87 mutató visszaolvasása
E751  85 86     STA   86
E753  60        RTS

******************************  a karakter beolvasása
E754  C9 20     CMP   #20       " " üresjel
E756  B0 0B     BCS   E763      ha nagyobb, beírni a pufferbe!
E758  AA        TAX             a kód tárolása
E759  A9 20     LDA   #20       üresjel
E75B  91 A5     STA   (A5),Y    felírása a pufferbe
E75D  C8        INY             puffer mutató növelése
E75E  8A        TXA             kód az akku-ban
E75F  20 06 E7  JSR   E706      hozzátartozó szöveg beolvasása
E762  60        RTS

E763  91 A5     STA   (A5),Y    karakter felírása a pufferbe
E765  C8        INY             és a mutató növelése
E766  60        RTS

******************************  a hibaüzenet egy karakterének olvasása
E767  E6 86     INC   86
E769  D0 02     BNE   E76D      mutató növelése
E76B  E6 87     INC   87
E76D  A1 86     LDA   (86,X)    karakter beolvasása
E76F  0A        ASL   A         7. bit a Carry-be
E770  A1 86     LDA   (86,X)    a karakter beolvasása
E772  29 7F     AND   #7F       a 7. bit törlése
E774  60        RTS
```

```
******************************          a mutató növelése
E775  20 6D E7    JSR   E76D            7. bit a Carry-be
E778  E6 86       INC   86
E77A  D0 02       BNE   E77E            a mutató növelése
E77C  E6 87       INC   87
E77E  60          RTS

******************************
E77F  60          RTS

******************************
E780  60          RTS

E781  EA          NOP
E782  EA          NOP
E783  EA          NOP
E784  EA          NOP
E785  EA          NOP
E786  EA          NOP
E787  EA          NOP
E788  EA          NOP
E789  EA          NOP
E78A  EA          NOP
E78B  EA          NOP
E78C  EA          NOP
E78D  EA          NOP
E78E  EA          NOP
E78F  EA          NOP
E790  EA          NOP
E791  EA          NOP
E792  EA          NOP
E793  EA          NOP
E794  EA          NOP
E795  EA          NOP
E796  EA          NOP
E797  EA          NOP
E798  EA          NOP
E799  EA          NOP
E79A  EA          NOP
E79B  EA          NOP
E79C  EA          NOP
E79D  EA          NOP
E79E  EA          NOP
E79F  EA          NOP
E7A0  EA          NOP
E7A1  EA          NOP
E7A2  60          RTS

******************************          & - utasítás
E7A3  A9 8D       LDA   #8D
E7A5  20 68 C2    JSR   C268            utasítás-sor ellenőrzése
E7A8  20 58 F2    JSR   F258            /RTS/
E7AB  AD 78 02    LDA   0278            adatnevek számának
E7AE  48          PHA                   tárolása
E7AF  A9 01       LDA   #01
E7B1  8D 78 02    STA   0278            egy adatnév
E7B4  A9 FF       LDA   #FF
E7B6  85 86       STA   86
E7B8  20 4F C4    JSR   C44F            file keresése
E7BB  AD 80 02    LDA   0280
E7BE  D0 05       BNE   E7C5            megvan?
E7C0  A9 39       LDA   #39
E7C2  20 C8 C1    JSR   C1C8            39, "file not found"
E7C5  68          PLA
E7C6  8D 78 02    STA   0278            file-nevek számának visszaolvasása
E7C9  AD 80 02    LDA   0280
E7CC  85 80       STA   80              sáv
```

```
E7CE  AD 85 02  LDA  0285
E7D1  85 81     STA  81       és szektor
E7D3  A9 03     LDA  #03      "USR" file-tipus
E7D5  20 77 D4  JSR  D477     a puffer lefoglalása, első blokk ovasása
E7D8  A9 00     LDA  #00
E7DA  85 87     STA  87       az ellenőrzőösszeg törlése
E7DC  20 39 E8  JSR  E839     egy byte beolvasása a file-ból
E7DF  85 88     STA  88       tárolása, kezdőcím /Lo/
E7E1  20 4B E8  JSR  E84B
E7E4  20 39 E8  JSR  E839     a következő byte beolvasása a file-ból
E7E7  85 89     STA  89       kezdőcím /Hi/
E7E9  20 4B E8  JSR  E84B     ellenőrzőösszeg kialakítása
E7EC  A5 86     LDA  86
E7EE  F0 0A     BEQ  E7FA
E7F0  A5 88     LDA  88
E7F2  48        PHA           a program indítási címének tárolása
E7F3  A5 89     LDA  89
E7F5  48        PHA
E7F6  A9 00     LDA  #00
E7F8  85 86     STA  86
E7FA  20 39 E8  JSR  E839     egy byte beolvasása a file-ból
E7FD  85 8A     STA  8A       tárolása
E7FF  20 4B E8  JSR  E84B     ellenőrzőösszeg képzése
E802  20 39 E8  JSR  E839     egy byte beolvasása a file-ból
E805  A0 00     LDY  #00
E807  91 88     STA  (88),Y   tárolása program-byte-ként
E809  20 4B E8  JSR  E84B     ellenőrzőösszeg képzése
E80C  A5 88     LDA  88
E80E  18        CLC
E80F  69 01     ADC  #01
E811  85 88     STA  88       a $88/$89 mutatók növelése
E813  90 02     BCC  E817
E815  E6 89     INC  89
E817  C6 8A     DEC  8A       a számláló csökkentése
E819  D0 E7     BNE  E802
E81B  20 35 CA  JSR  CA35     a következő byte beolvasása
E81E  A5 85     LDA  85       adatbyte
E820  C5 87     CMP  87       egyenlő az ellenőrzőösszeggel?
E822  F0 08     BEQ  E82C     igen
E824  20 3E DE  JSR  DE3E     paraméter átadása a lemezvezérlőnek
E827  A9 50     LDA  #50
E829  20 45 E6  JSR  E645     50, "record not present"
E82C  A5 F8     LDA  F8       vége?
E82E  D0 A8     BNE  E7D8     nem, következő blokk
E830  68        PLA
E831  85 89     STA  89
E833  68        PLA           visszatölteni a program kezdési címét!
E834  85 88     STA  88
E836  6C 88 00  JMP  (0088)   és a program futtatása
E839  20 35 CA  JSR  CA35     egy byte beolvasása a file-ból
E83C  A5 F8     LDA  F8       vége?
E83E  D0 08     BNE  E848     nem
E840  20 3E DE  JSR  DE3E     a paraméter átadása a lemezvezérlőnek
E843  A9 51     LDA  #51
E845  20 45 E6  JSR  E645     51, "overflow in record"
E848  A5 85     LDA  85       adatbyte
E84A  60        RTS

******************************  ellenőrzőösszeg képzése
E84B  18        CLC
E84C  65 87     ADC  87
E84E  69 00     ADC  #00
E850  85 87     STA  87
E852  60        RTS
```

```
******************************        a soros busz IRQ-rutinja
E853  AD 01 18   LDA   1801           az A-kapu olvasása, IRQ-kapcsoló törlése
E856  A9 01      LDA   #01            az "ATN vétel"-kapcsoló beállítása
E858  85 7C      STA   7C
E85A  60         RTS

******************************        soros busz kezelése
E85B  78         SEI
E85C  A9 00      LDA   #00
E85E  85 7C      STA   7C             az "ATN vétel"-kapcsoló törlése
E860  85 79      STA   79             a LISTEN-kapcsoló törlése
E862  85 7A      STA   7A             a TALK-kapcsoló törlése
E864  A2 45      LDX   #45
E866  9A         TXS                  verem-mutató inicializálása
E867  A9 80      LDA   #80
E869  85 F8      STA   F8             a vég-kapcsoló törlése
E86B  85 7D      STA   7D             az EOI-kapcsoló törlése
E86D  20 B7 E9   JSR   E9B7           CLOCK OUT /Lo/
E870  20 A5 E9   JSR   E9A5           DATA OUT, Bit "Ø" /Hi/
E873  AD 00 18   LDA   1800
E876  09 10      ORA   #10            adatvonalak inputra kapcsolása
E878  8D 00 18   STA   1800
E87B  AD 00 18   LDA   1800           az IEC-kapu olvasása
E87E  10 57      BPL   E8D7           EOI?
E880  29 04      AND   #04            CLOCK IN?
E882  D0 F7      BNE   E87B           nem
E884  20 C9 E9   JSR   E9C9           a byte beolvasása a buszról
E887  C9 3F      CMP   #3F            Unlisten?
E889  D0 06      BNE   E891           nem
E88B  A9 00      LDA   #00
E88D  85 79      STA   79             a LISTEN-kapcsoló visszaállítása
E88F  F0 71      BEQ   E902
E891  C9 5F      CMP   #5F            Untalk?
E893  D0 06      BNE   E89B           nem
E895  A9 00      LDA   #00
E897  85 7A      STA   7A             a TALK-kapcsoló visszaállítása
E899  F0 67      BEQ   E902
E89B  C5 78      CMP   78             TALK-cím?
E89D  D0 0A      BNE   E8A9           nem
E89F  A9 01      LDA   #01
E8A1  85 7A      STA   7A             a TALK-kapcsoló beállítása
E8A3  A9 00      LDA   #00
E8A5  85 79      STA   79             a LISTEN-kapcsoló visszaállítása
E8A7  F0 29      BEQ   E8D2
E8A9  C5 77      CMP   77             LISTEN-cím?
E8AB  D0 0A      BNE   E8B7           nem
E8AD  A9 01      LDA   #01
E8AF  85 79      STA   79             a LISTEN-kapcsoló beállítása
E8B1  A9 00      LDA   #00
E8B3  85 7A      STA   7A             a TALK-kapcsoló visszaállítása
E8B5  F0 1B      BEQ   E8D2
E8B7  AA         TAX
E8B8  29 60      AND   #60
E8BA  C9 60      CMP   #60            az 5. és 6. bit magas?
E8BC  D0 3F      BNE   E8FD           nem
E8BE  8A         TXA
E8BF  85 84      STA   84             másodlagos cím
E8C1  29 0F      AND   #0F
E8C3  85 83      STA   83             csatorna-szám
E8C5  A5 84      LDA   84
E8C7  29 F0      AND   #F0
E8C9  C9 E0      CMP   #E0            CLOSE?
E8CB  D0 35      BNE   E902
E8CD  58         CLI
E8CE  20 C0 DA   JSR   DAC0           CLOSE-rutin
E8D1  78         SEI
E8D2  2C 00 18   BIT   1800
```

```
E8D5  30 AD      BMI   E884
E8D7  A9 00      LDA   #00
E8D9  85 7D      STA   7D      az EOI beállítása
E8DB  AD 00 18   LDA   1800    IEC-kapu
E8DE  29 EF      AND   #EF     az adatvonalak outputra kapcsolása
E8E0  8D 00 18   STA   1800
E8E3  A5 79      LDA   79      LISTEN aktiv?
E8E5  F0 06      BEQ   E8ED    nem
E8E7  20 2E EA   JSR   EA2E    adatok vétele
E8EA  4C E7 EB   JMP   EBE7    ugrás a sorbanállási ciklusra

E8ED  A5 7A      LDA   7A      TALK aktiv?
E8EF  F0 09      BEQ   E8FA    nem
E8F1  20 9C E9   JSR   E99C    DATA OUT, bit "1" /Lo/
E8F4  20 AE E9   JSR   E9AE    CLOCK OUT /Hi/
E8F7  20 09 E9   JSR   E909    adatok küldése
E8FA  4C 4E EA   JMP   EA4E    ugrás a sorbanállási ciklushoz

E8FD  A9 10      LDA   #10     a byte figyelmen kivűl marad! /TALK, LISTEN/
E8FF  8D 00 18   STA   1800    adatvonalak inputra kapcsolása
E902  2C 00 18   BIT   1800
E905  10 D0      BPL   E8D7
E907  30 F9      BMI   E902    handshake-t megvárni!

******************************         adatok küldése
E909  78         SEI
E90A  20 EB D0   JSR   D0EB    olvasás-csatorna megnyitása
E90D  B0 06      BCS   E915    a csatorna aktív?
E90F  A6 82      LDX   82      csatorna-szám
E911  B5 F2      LDA   F2,X    a READ-kapcsoló magas?
E913  30 01      BMI   E916    igen
E915  60         RTS

E916  20 59 EA   JSR   EA59    ellenőrzés az EOI szempontjából
E919  20 C0 E9   JSR   E9C0    IEC-kapu olvasása
E91C  29 01      AND   #01     adatbit leválasztása
E91E  08         PHP           és tárolása
E91F  20 B7 E9   JSR   E9B7    CLOCK OUT /Lo/
E922  28         PLP
E923  F0 12      BEQ   E937
E925  20 59 EA   JSR   EA59    ellenőrzés az EOI szempontjából
E928  20 C0 E9   JSR   E9C0    az IEC-kapu olvasása
E92B  29 01      AND   #01     adatbit leválasztása
E92D  D0 F6      BNE   E925
E92F  A6 82      LDX   82      csatorna-szám
E931  B5 F2      LDA   F2,X
E933  29 08      AND   #08
E935  D0 14      BNE   E94B
E937  20 59 EA   JSR   EA59    ellenőrzés az EOI szempontjából
E93A  20 C0 E9   JSR   E9C0    az IEC-kapu olvasása
E93D  29 01      AND   #01     adatbit leválasztása
E93F  D0 F6      BNE   E937
E941  20 59 EA   JSR   EA59    ellenőrzés az EOI szempontjából
E944  20 C0 E9   JSR   E9C0    az IEC-kapu olvasása
E947  29 01      AND   #01     adatbit leválasztása
E949  F0 F6      BEQ   E941
E94B  20 AE E9   JSR   E9AE    CLOCK OUT /Hi/
E94E  20 59 EA   JSR   EA59    ellenőrzés az EOI szempontjából
E951  20 C0 E9   JSR   E9C0    az IEC-kapu olvasása
E954  29 01      AND   #01     adatbit leválasztása
E956  D0 F3      BNE   E94B
E958  A9 08      LDA   #08     számláló 8 biten soros átvitelhez
E95A  85 98      STA   98      az IEC-kapu olvasása
E95C  20 C0 E9   JSR   E9C0    az adatbit leválasztása
E95F  29 01      AND   #01
E961  D0 36      BNE   E999
E963  A6 82      LDX   82
```

```
E965  BD 3E 02    LDA   023E,X
E968  6A          ROR   A           legalsó bit a Carry-be
E969  9D 3E 02    STA   023E,X
E96C  B0 05       BCS   E973        bit magas
E96E  20 A5 E9    JSR   E9A5        DATA OUT, "Ø" bitet kihozni!
E971  D0 03       BNE   E976        feltétlen ugrás
E973  20 9C E9    JSR   E99C        DATA OUT, "1" bitet kihozni!
E976  20 B7 E9    JSR   E9B7        a CLOCK OUT beállítása!
E979  A5 23       LDA   23
E97B  D0 03       BNE   E980
E97D  20 F3 FE    JSR   FEF3        a soros busz késleltetése
E980  20 FB FE    JSR   FEFB        a DATA OUT és a CLOCK OUT beállítása
E983  C6 98       DEC   98          minden bit átvitele megtörtént?
E985  D0 D5       BNE   E95C        nem
E987  20 59 EA    JSR   EA59        ellenőrzés az EOI szempontjából
E98A  20 C0 E9    JSR   E9C0        az IEC-kapu olvasása
E98D  29 01       AND   #01         az adatbit leválasztása
E98F  F0 F6       BEQ   E987
E991  58          CLI
E992  20 AA D3    JSR   D3AA        a következő adatbyte beolvasása
E995  78          SEI
E996  4C 0F E9    JMP   E90F        és kiírása

E999  4C 4E EA    JMP   EA4E        ugrás a sorbanállási ciklusra

******************************      DATA OUT /Lo/
E99C  AD 00 18    LDA   1800
E99F  29 FD       AND   #FD         az "1" bit kiírása
E9A1  8D 00 18    STA   1800
E9A4  60          RTS

******************************      DATA OUT /Hi/
E9A5  AD 00 18    LDA   1800
E9A8  09 02       ORA   #02         a "Ø" bit kiírása
E9AA  8D 00 18    STA   1800
E9AD  60          RTS

******************************      CLOCK OUT /Hi/
E9AE  AD 00 18    LDA   1800
E9B1  09 08       ORA   #08         a 3. bit beállítása
E9B3  8D 00 18    STA   1800
E9B6  60          RTS

******************************      CLOCK OUT /Lo/
E9B7  AD 00 18    LDA   1800
E9BA  29 F7       AND   #F7         a 3. bit törlése
E9BC  8D 00 18    STA   1800
E9BF  60          RTS

******************************      az IEC-kapu olvasása
E9C0  AD 00 18    LDA   1800
E9C3  CD 00 18    CMP   1800        várakozás
E9C6  D0 F8       BNE   E9C0
E9C8  60          RTS

******************************
E9C9  A9 08       LDA   #08
E9CB  85 98       STA   98          bit-számláló, soros output
E9CD  20 59 EA    JSR   EA59        ellenőrzés az EOI szempontjából
E9D0  20 C0 E9    JSR   E9C0        az IEC-kapu olvasása
E9D3  29 04       AND   #04         CLOCK IN?
E9D5  D0 F6       BNE   E9CD        nem, várni
E9D7  20 9C E9    JSR   E99C        DATA OUT, "1" bit
E9DA  A9 01       LDA   #01
E9DC  4C 20 FF    JMP   FF20        óra beállítása
```

```
E9DF  20 59 EA   JSR  EA59    ellenőrzés az EOI szempontjából
E9E2  AD 0D 18   LDA  180D
E9E5  29 40      AND  #40     lejárt az óra?
E9E7  D0 09      BNE  E9F2    igen, EOI
E9E9  20 C0 E9   JSR  E9C0    az IEC-kapu olvasása
E9EC  29 04      AND  #04     CLOCK IN?
E9EE  F0 EF      BEQ  E9DF    nem, várni
E9F0  D0 19      BNE  EA0B
E9F2  20 A5 E9   JSR  E9A5    DATA OUT "0" bit, /Hi/
E9F5  A2 0A      LDX  #0A     10
E9F7  CA         DEX          késleltetési ciklus kb. 50 µs.
E9F8  D0 FD      BNE  E9F7
E9FA  20 9C E9   JSR  E99C    DATA OUT, "1" bit /Lo/
E9FD  20 59 EA   JSR  EA59    ellenőrzés az EOI szempontjából
EA00  20 C0 E9   JSR  E9C0    az IEC-kapu olvasása
EA03  29 04      AND  #04     CLOCK IN?
EA05  F0 F6      BEQ  E9FD    nem, várni
EA07  A9 00      LDA  #00
EA09  85 F8      STA  F8      az EOI-kapcsoló beállítása
EA0B  AD 00 18   LDA  1800    IEC-kapu
EA0E  49 01      EOR  #01     az adatbit invertálása
EA10  4A         LSR  A
EA11  29 02      AND  #02
EA13  D0 F6      BNE  EA0B    CLOCK IN?
EA15  EA         NOP
EA16  EA         NOP
EA17  EA         NOP
EA18  66 85      ROR  85      a következő bit előkészítése
EA1A  20 59 EA   JSR  EA59    ellenőrzés az EOI szempontjából
EA1D  20 C0 E9   JSR  E9C0    az IEC-kapu olvasása
EA20  29 04      AND  #04     CLOCK IN?
EA22  F0 F6      BEQ  EA1A    nem
EA24  C6 98      DEC  98      a bitszámláló csökkentése
EA26  D0 E3      BNE  EA0B    minden bit átvitele megtörtént?
EA28  20 A5 E9   JSR  E9A5    DATA OUT, "0" bit /Hi/
EA2B  A5 85      LDA  85      adatbyte betöltése
EA2D  60         RTS

******************************    adatbeolvasás a soros buszról
EA2E  78         SEI
EA2F  20 07 D1   JSR  D107    íráscsatorna megnyitása
EA32  B0 05      BCS  EA39    nem aktiv a csatorna?
EA34  B5 F2      LDA  F2,X    WRITE-kapcsoló
EA36  6A         ROR  A
EA37  B0 0B      BCS  EA44    alacsony?
EA39  A5 84      LDA  84      másodlagos cím
EA3B  29 F0      AND  #F0
EA3D  C9 F0      CMP  #F0     OPEN-utasítás?
EA3F  F0 03      BEQ  EA44    igen
EA41  4C 4E EA   JMP  EA4E    várakozási ciklushoz

EA44  20 C9 E9   JSR  E9C9    a soros buszról az adatbyte beolvasása
EA47  58         CLI
EA48  20 B7 CF   JSR  CFB7    és beírása a pufferbe
EA4B  4C 2E EA   JMP  EA2E    a ciklus elejéhez

EA4E  A9 00      LDA  #00
EA50  8D 00 18   STA  1800    az IEC-kapu visszaállítása
EA53  4C E7 EB   JMP  EBE7    várakozási ciklushoz

EA56  4C 5B E8   JMP  E85B    soros busz - főciklusához

******************************
EA59  A5 7D      LDA  7D      megtörtént az EOI vétele?
EA5B  F0 06      BEQ  EA63    igen
EA5D  AD 00 18   LDA  1800    IEC-kapu
EA60  10 09      BPL  EA6B
EA62  60         RTS
```

```
EA63  AD 00 18     LDA   1800      IEC-kapu
EA66  10 FA        BPL   EA62
EA68  4C 5B E8     JMP   E85B      a soros busz főciklusához

EA6B  4C D7 E8     JMP   E8D7      az EOI beállítása

****************************       hardver-hibáknál a LED-ek villognak, önteszt
EA6E  A2 00        LDX   #00       egyszeri villogás, Zeropage
EA70  2C          .BYTE  $2C
EA71  A6 6F        LDX   6F        X+1-szeri villogás, RAM/ROM hiba
EA73  9A           TXS
EA74  BA           TSX
EA75  A9 08        LDA   #08       a LED-bit a kapun
EA77  0D 00 1C     ORA   1C00
EA7A  4C EA FE     JMP   FEEA      LED kikapcsolása, vissza $EA7D-re

EA7D  98           TYA
EA7E  18           CLC
EA7F  69 01        ADC   #01
EA81  D0 FC        BNE   EA7F
EA83  88           DEY
EA84  D0 F8        BNE   EA7E
EA86  AD 00 1C     LDA   1C00
EA89  29 F7        AND   #F7       LED kikapcsolása
EA8B  8D 00 1C     STA   1C00
EA8E  98           TYA
EA8F  18           CLC
EA90  69 01        ADC   #01
EA92  D0 FC        BNE   EA90      késleltetési ciklus
EA94  88           DEY
EA95  D0 F8        BNE   EA8F
EA97  CA           DEX
EA98  10 DB        BPL   EA75
EA9A  E0 FC        CPX   #FC
EA9C  D0 F0        BNE   EA8E      késleltetést megvárni!
EA9E  F0 D4        BEQ   EA74      LED ismételt bekapcsolása

****************************       RESET-rutin
EAA0  78           SEI
EAA1  D8           CLD
EAA2  A2 FF        LDX   #FF
EAA4  4C 10 FF     JMP   FF10      ugrás a rutinra

EAA7  E8           INX
EAA8  A0 00        LDY   #00
EAAA  A2 00        LDX   #00
EAAC  8A           TXA
EAAD  95 00        STA   00,X      zeropage törlése
EAAF  E8           INX
EAB0  D0 FA        BNE   EAAC
EAB2  8A           TXA
EAB3  D5 00        CMP   00,X      a byte törölve?
EAB5  D0 B7        BNE   EA6E      ha nem, hibajelzéshez /villogás/!
EAB7  F6 00        INC   00,X
EAB9  C8           INY
EABA  D0 FB        BNE   EAB7
EABC  D5 00        CMP   00,X
EABE  D0 AE        BNE   EA6E      hiba
EAC0  94 00        STY   00,X
EAC2  B5 00        LDA   00,X
EAC4  D0 A8        BNE   EA6E      hiba
EAC6  E8           INX
EAC7  D0 E9        BNE   EAB2
EAC9  E6 6F        INC   6F
EACB  86 76        STX   76
EACD  A9 00        LDA   #00
EACF  85 75        STA   75
```

```
EAD1  A8        TAY
EAD2  A2 20     LDX  #20        32. lap tesztelése
EAD4  18        CLC
EAD5  C6 76     DEC   76
EAD7  71 75     ADC  (75),Y
EAD9  C8        INY
EADA  D0 FB     BNE   EAD7
EADC  CA        DEX
EADD  D0 F6     BNE   EAD5      ROM tesztelése
EADF  69 00     ADC  #00
EAE1  AA        TAX
EAE2  C5 76     CMP   76
EAE4  D0 39     BNE   EB1F      ROM-hiba
EAE6  E0 C0     CPX  #C0
EAE8  D0 DF     BNE   EAC9
EAEA  A9 01     LDA  #01
EAEC  85 76     STA   76
EAEE  E6 6F     INC   6F
EAF0  A2 07     LDX  #07        RAM tesztelése a 7. laptól kezdve
EAF2  98        TYA
EAF3  18        CLC
EAF4  65 76     ADC   76
EAF6  91 75     STA  (75),Y
EAF8  C8        INY
EAF9  D0 F7     BNE   EAF2
EAFB  E6 76     INC   76
EAFD  CA        DEX
EAFE  D0 F2     BNE   EAF2
EB00  A2 07     LDX  #07
EB02  C6 76     DEC   76
EB04  88        DEY
EB05  98        TYA
EB06  18        CLC
EB07  65 76     ADC   76
EB09  D1 75     CMP  (75),Y
EB0B  D0 12     BNE   EB1F      RAM-hiba
EB0D  49 FF     EOR  #FF
EB0F  91 75     STA  (75),Y
EB11  51 75     EOR  (75),Y
EB13  91 75     STA  (75),Y
EB15  D0 08     BNE   EB1F      RAM-hiba
EB17  98        TYA
EB18  D0 EA     BNE   EB04
EB1A  CA        DEX
EB1B  D0 E5     BNE   EB02      tesztelést folytatni!
EB1D  F0 03     BEQ   EB22      ok
EB1F  4C 71 EA  JMP   EA71      hibajelzéshez!

EB22  A2 45     LDX  #45
EB24  9A        TXS             a verem-mutató inicializálása
EB25  AD 00 1C  LDA   1C00
EB28  29 F7     AND  #F7        LED kikapcsolása
EB2A  8D 00 1C  STA   1C00
EB2D  A9 01     LDA  #01
EB2F  8D 0C 18  STA   180C      CA1 /ATN-IN/ átbillentése
EB32  A9 82     LDA  #82
EB34  8D 0D 18  STA   180D      ATN-IN általi megszakítás engedélyezése
EB37  8D 0E 18  STA   180E
EB3A  AD 00 18  LDA   1800      a B-kapu olvasása
EB3D  29 60     AND  #60        az 5. és 6. bit leválasztása
EB3F  0A        ASL   A
EB40  2A        ROL   A
EB41  2A        ROL   A         léptetés a Ø. és 1. bitpozicióba
EB42  2A        ROL   A
EB43  09 48     ORA  #48        offset-8 és Talk $4Ø összeadása
EB45  85 78     STA   78        TALK - egységszám
EB47  49 60     EOR  #60        6. bit törlése és az 5. bit beállitása
```

```
EB49  85 77      STA  77      egységszám plusz $20 LISTEN-hez
EB4B  A2 00      LDX  #00
EB4D  A0 00      LDY  #00
EB4F  A9 00      LDA  #00
EB51  95 99      STA  99,X    puffer-címek alsó byte-ja
EB53  E8         INX
EB54  B9 E0 FE   LDA  FEE0,Y  felső byte a táblázatból
EB57  95 99      STA  99,X    tárolás
EB59  E8         INX
EB5A  C8         INY
EB5B  C0 05      CPY  #05     5-ös puffertár
EB5D  D0 F0      BNE  EB4F
EB5F  A9 00      LDA  #00
EB61  95 99      STA  99,X
EB63  E8         INX          $A3/$A4 mutató $200-on, input-puffer
EB64  A9 02      LDA  #02
EB66  95 99      STA  99,X
EB68  E8         INX
EB69  A9 D5      LDA  #D5
EB6B  95 99      STA  99,X
EB6D  E8         INX          $A5/$A6 mutató $2D5-ön, hibaüzenet-puffer
EB6E  A9 02      LDA  #02
EB70  95 99      STA  99,X
EB72  A9 FF      LDA  #FF
EB74  A2 12      LDX  #12
EB76  9D 2B 02   STA  022B,X  csatorna-táblázat törlése $FF-el
EB79  CA         DEX
EB7A  10 FA      BPL  EB76
EB7C  A2 05      LDX  #05
EB7E  95 A7      STA  A7,X    a puffer-táblázatok törlése
EB80  95 AE      STA  AE,X
EB82  95 CD      STA  CD,X    az oldal-szektor táblázat törlése
EB84  CA         DEX
EB85  10 F7      BPL  EB7E
EB87  A9 05      LDA  #05     5. puffer
EB89  85 AB      STA  AB      a 4. csatorna hozzárendelése
EB8B  A9 06      LDA  #06     6. puffer
EB8D  85 AC      STA  AC      az 5. csatorna hozzárendelés
EB8F  A9 FF      LDA  #FF
EB91  85 AD      STA  AD
EB93  85 B4      STA  B4
EB95  A9 05      LDA  #05
EB97  8D 3B 02   STA  023B    5. csatorna WRITEkapcsoló törlése
EB9A  A9 84      LDA  #84
EB9C  8D 3A 02   STA  023A    4. csatorna WRITEkapcsoló beállítása
EB9F  A9 0F      LDA  #0F     csatornafoglaltsági-regiszter inicializálása
EBA1  8D 56 02   STA  0256    ha a bit értéke "1" - szabad csatorna!
EBA4  A9 01      LDA  #01
EBA6  85 F6      STA  F6      WRITE-kapcsoló
EBA8  A9 88      LDA  #88
EBAA  85 F7      STA  F7      READ-kapcsoló
EBAC  A9 E0      LDA  #E0     az 5. puffer szabad
EBAE  8D 4F 02   STA  024F    a pufferfoglaltsági-regiszter inicializálása
EBB1  A9 FF      LDA  #FF     $24F/$250, 16 bit, "1" - puffer foglalt
EBB3  8D 50 02   STA  0250
EBB6  A9 01      LDA  #01
EBB8  85 1C      STA  1C      "Write Protect" kapcsolók
EBBA  85 1D      STA  1D
EBBC  20 63 CB   JSR  CB63    UO-vektor beállítása
EBBF  20 FA CE   JSR  CEFA    a csatorna-táblázat inicializálása
EBC2  20 59 F2   JSR  F259    lemezvezérlő inicializálása
EBC5  A9 22      LDA  #22
EBC7  85 65      STA  65
EBC9  A9 EB      LDA  #EB     $65/$66 mutató $EB22-n
EBCB  85 66      STA  66
EBCD  A9 0A      LDA  #0A
EBCF  85 69      STA  69      10-es lépéstávolság szektor-hozzárendeléssel
```

```
EBD1   A9 05       LDA   #05
EBD3   85 6A       STA    6A         5 olvasási kisérlet
EBD5   A9 73       LDA   #73         a bekapcsolási üzenet előkészítése
EBD7   20 C1 E6    JSR    E6C1       73, "cbm dos v2.6 1541"
EBDA   A9 00       LDA   #00         1., 3. és 4. bit kimeneten
EBDC   8D 00 18    STA    1800       az adatirány: B-kapu
EBDF   A9 1A       LDA   #1A
EBE1   8D 02 18    STA    1802       az adatregiszter törlése
EBE4   20 80 E7    JSR    E780       automatikus start ellenőrzése
EBE7   58          CLI
EBE8   AD 00 18    LDA    1800
EBEB   29 E5       AND   #E5         a soros kapu visszaállítása
EBED   8D 00 18    STA    1800
EBF0   AD 55 02    LDA    0255       az utasításkapcsoló magas?
EBF3   F0 0A       BEQ    EBFF       nem
EBF5   A9 00       LDA   #00
EBF7   8D 55 02    STA    0255       az utasításkapcsoló visszaállítása
EBFA   85 67       STA    67
EBFC   20 46 C1    JSR    C146       utasítás elemzése és végrehajtása

****************************         várakozási ciklus
EBFF   58          CLI
EC00   A5 7C       LDA    7C         megtörtént az ATN-jel észlelése?
EC02   F0 03       BEQ    EC07       nem
EC04   4C 5B E8    JMP    E85B       IEC-rutinhoz

EC07   58          CLI
EC08   A9 0E       LDA   #0E         14
EC0A   85 72       STA    72         mint másodlagos cím
EC0C   A9 00       LDA   #00
EC0E   85 6F       STA    6F         job-számláló
EC10   85 70       STA    70
EC12   A6 72       LDX    72
EC14   BD 2B 02    LDA    022B,X     másodlagos cím
EC17   C9 FF       CMP   #FF         hozzá van rendelve a csatorna?
EC19   F0 10       BEQ    EC2B       nincs
EC1B   29 3F       AND   #3F
EC1D   85 82       STA    82         csatorna-szám
EC1F   20 93 DF    JSR    DF93       a puffer-szám beolvasása
EC22   AA          TAX
EC23   BD 5B 02    LDA    025B,X     ègységszám
EC26   29 01       AND   #01
EC28   AA          TAX
EC29   F6 6F       INC    6F,X       a job-számláló növelése
EC2B   C6 72       DEC    72         cím /Lo/
EC2D   10 E3       BPL    EC12       keresés
EC2F   A0 04       LDY   #04         puffer-számláló
EC31   B9 00 00    LDA    0000,Y     müködik a lemezvezérlő
EC34   10 05       BPL    EC3B       nincs
EC36   29 01       AND   #01         egységszám leválasztása
EC38   AA          TAX
EC39   F6 6F       INC    6F,X       a job-számláló növelése
EC3B   88          DEY
EC3C   10 F3       BPL    EC31       következő puffer
EC3E   78          SEI
EC3F   AD 00 1C    LDA    1C00
EC42   29 F7       AND   #F7         a LED-bit törlése
EC44   48          PHA
EC45   A5 7F       LDA    7F         egységszám
EC47   85 86       STA    86
EC49   A9 00       LDA   #00
EC4B   85 7F       STA    7F         Ø-s meghajtó
EC4D   A5 6F       LDA    6F         Ø-s meghajtó job-ja?
EC4F   F0 0B       BEQ    EC5C       nem
EC51   A5 1C       LDA    1C         Ø-s meghajtó "Write Protect"?
EC53   F0 03       BEQ    EC58       nem
EC55   20 13 D3    JSR    D313       Ø-s meghajtóhoz irányuló összes csatorna le-
```

```
EC58   68           PLA                 zárása
EC59   09 08        ORA    #08          a LED-bit beállítása
EC5B   48           PHA
EC5C   E6 7F        INC    7F           a meghajtó-szám növelése
EC5E   A5 70        LDA    70           1. meghajtó job-ja?
EC60   F0 0B        BEQ    EC6D         nem
EC62   A5 1D        LDA    1D           1. meghajtó írásvédelme?
EC64   F0 03        BEQ    EC69         nem
EC66   20 13 D3     JSR    D313         1. meghajtóhoz irányuló összes csato
EC69   68           PLA                 zárása
EC6A   09 00        ORA    #00
EC6C   48           PHA
EC6D   A5 86        LDA    86
EC6F   85 7F        STA    7F           az egységszám visszaolvasása
EC71   68           PLA                 LED-bit
EC72   AE 6C 02     LDX    026C         megszakítás-számláló
EC75   F0 21        BEQ    EC98         nullán?
EC77   AD 00 1C     LDA    1C00
EC7A   E0 80        CPX    #80
EC7C   D0 03        BNE    EC81
EC7E   4C 8B EC     JMP    EC8B

EC81   AE 05 18     LDX    1805         óra-megszakítás törlése
EC84   30 12        BMI    EC98
EC86   A2 A0        LDX    #A0
EC88   8E 05 18     STX    1805         óra beállítása
EC8B   CE 6C 02     DEC    026C         a számláló csökkentése
EC8E   D0 08        BNE    EC98         még nem nulla?
EC90   4D 6D 02     EOR    026D
EC93   A2 10        LDX    #10
EC95   8E 6C 02     STX    026C         újra beállítani a számlálót!
EC98   8D 00 1C     STA    1C00         LED bekapcsolása/kikapcsolása
EC9B   4C FF EB     JMP    EBFF         vissza a várakozási ciklushoz!

******************************          LOAD "$"
EC9E   A9 00        LDA    #00
ECA0   85 83        STA    83           másodlagos cím nulla
ECA2   A9 01        LDA    #01
ECA4   20 E2 D1     JSR    D1E2         csatornát és puffert keresni!
ECA7   A9 00        LDA    #00
ECA9   20 C8 D4     JSR    D4C8         a puffer mutató inicializálása
ECAC   A6 82        LDX    82           csatorna-szám
ECAE   A9 00        LDA    #00
ECB0   9D 44 02     STA    0244,X       a mutató a végén nullával egyenlő
ECB3   20 93 DF     JSR    DF93         a puffer-szám beolvasása
ECB6   AA           TAX
ECB7   A5 7F        LDA    7F           egységszám
ECB9   9D 5B 02     STA    025B,X       beírása a táblázatba
ECBC   A9 01        LDA    #01          1
ECBE   20 F1 CF     JSR    CFF1         beírni a pufferbe!
ECC1   A9 04        LDA    #04          4, $0401 kezdőcímet
ECC3   20 F1 CF     JSR    CFF1         beírni a pufferbe!
ECC6   A9 01        LDA    #01          2 x 1
ECC8   20 F1 CF     JSR    CFF1
ECCB   20 F1 CF     JSR    CFF1         kapcsolócímként beírni a pufferbe
ECCE   AD 72 02     LDA    0272         egységszámot
ECD1   20 F1 CF     JSR    CFF1         sor-számként beírni a pufferbe!
ECD4   A9 00        LDA    #00          sor-szám /Hi/
ECD6   20 F1 CF     JSR    CFF1         a pufferben
ECD9   20 59 ED     JSR    ED59         bejegyzés a pufferben
ECDC   20 93 DF     JSR    DF93         a puffer-szám beolvasása
ECDF   0A           ASL    A
ECE0   AA           TAX
ECE1   D6 99        DEC    99,X         a puffermutató csökkentése
ECE3   D6 99        DEC    99,X
ECE5   A9 00        LDA    #00
ECE7   20 F1 CF     JSR    CFF1         0, mint sorvég a pufferben
```

```
ECEA  A9 01      LDA  #01
ECEC  20 F1 CF   JSR  CFF1      2 x 1, mint kapcsolócím
ECEF  20 F1 CF   JSR  CFF1
ECF2  20 CE C6   JSR  C6CE      bejegyzés a pufferben
ECF5  90 2C      BCC  ED23      további bejegyzés?
ECF7  AD 72 02   LDA  0272      blokkszám /Lo/
ECFA  20 F1 CF   JSR  CFF1      a pufferben
ECFD  AD 73 02   LDA  0273      blokkszám /Hi/
ED00  20 F1 CF   JSR  CFF1      a pufferben
ED03  20 59 ED   JSR  ED59      bejegyzés a pufferben
ED06  A9 00      LDA  #00
ED08  20 F1 CF   JSR  CFF1      nulla, mint végjelölés a pufferben
ED0B  D0 DD      BNE  ECEA      megtelt a puffer? nem
ED0D  20 93 DF   JSR  DF93      a puffer-szám beolvasása
ED10  0A         ASL  A
ED11  AA         TAX
ED12  A9 00      LDA  #00
ED14  95 99      STA  99,X      puffer mutató nullán
ED16  A9 88      LDA  #88       a READ-kapcsoló beállítása
ED18  A4 82      LDY  82        csatorna-szám
ED1A  8D 54 02   STA  0254
ED1D  99 F2 00   STA  00F2,Y    csatorna-kapcsoló
ED20  A5 85      LDA  85        adatbyte
ED22  60         RTS

******************************
ED23  AD 72 02   LDA  0272      a blokkszám /Lo/
ED26  20 F1 CF   JSR  CFF1      beírása a pufferbe!
ED29  AD 73 02   LDA  0273      a blokkszám /Hi/
ED2C  20 F1 CF   JSR  CFF1      beírása a pufferbe
ED2F  20 59 ED   JSR  ED59      "Blocks free." a pufferben
ED32  20 93 DF   JSR  DF93      puffer-szám beolvasása
ED35  0A         ASL  A         2-szer
ED36  AA         TAX
ED37  D6 99      DEC  99,X
ED39  D6 99      DEC  99,X      puffer mutató minusz 2
ED3B  A9 00      LDA  #00
ED3D  20 F1 CF   JSR  CFF1
ED40  20 F1 CF   JSR  CFF1      háromszor nulla, mint programvég
ED43  20 F1 CF   JSR  CFF1
ED46  20 93 DF   JSR  DF93      a puffer-szám beolvasása
ED49  0A         ASL  A         2-szer
ED4A  A8         TAY
ED4B  B9 99 00   LDA  0099,Y    puffer mutató,
ED4E  A6 82      LDX  82
ED50  9D 44 02   STA  0244,X    mint végjel
ED53  DE 44 02   DEC  0244,X
ED56  4C 0D ED   JMP  ED0D

******************************  a tartalomjegyzék-sor átvitele
ED59  A0 00      LDY  #00
ED5B  B9 B1 02   LDA  02B1,Y    egy karakter a pufferból
ED5E  20 F1 CF   JSR  CFF1      az output-pufferbe
ED61  C8         INY
ED62  C0 1B      CPY  #1B       már 27 karakter?
ED64  D0 F5      BNE  ED5B
ED66  60         RTS

******************************  egy byte beolvasása a pufferból
ED67  20 37 D1   JSR  D137      egy byte beolvasása a pufferból
ED6A  F0 01      BEQ  ED6D      puffer mutató nulla?
ED6C  60         RTS

ED6D  85 85      STA  85        adatbyte tárolása
ED6F  A4 82      LDY  82        csatorna-szám
ED71  B9 44 02   LDA  0244,Y    végjel beállítása
ED74  F0 08      BEQ  ED7E      nulla /LOAD $/?
```

```
ED76   A9 80      LDA  #80
ED78   99 F2 00   STA  00F2,Y    READ-kapcsoló beállítása
ED7B   A5 85      LDA  85        adatbyte
ED7D   60         RTS

ED7E   48         PHA
ED7F   20 EA EC   JSR  ECEA      a tartalomjegyzék-sor előállítása a pufferben
ED82   68         PLA
ED83   60         RTS

****************************     V-utasítás, "Collect"
ED84   20 D1 C1   JSR  C1D1      egységszám megkeresése az input-sorban
ED87   20 42 D0   JSR  D042      a BAM betöltése
ED8A   A9 40      LDA  #40
ED8C   8D F9 02   STA  02F9
ED8F   20 B7 EE   JSR  EEB7      új BAM előállítása a pufferben
ED92   A9 00      LDA  #00
ED94   8D 92 02   STA  0292      a tart.j.betöltése, első bejegyzés megkeresése
ED97   20 AC C5   JSR  C5AC      megvan?
ED9A   D0 3D      BNE  EDD9
ED9C   A9 00      LDA  #00
ED9E   85 81      STA  81        Ø. szektor
EDA0   AD 85 FE   LDA  FE85      18
EDA3   85 80      STA  80        BAM 18-as sávja
EDA5   20 E5 ED   JSR  EDE5      tartalomjegyzék-blokkok lefoglalása
EDA8   A9 00      LDA  #00
EDAA   8D F9 02   STA  02F9
EDAD   20 FF EE   JSR  EEFF      BAM visszaírása a lemezre
EDB0   4C 94 C1   JMP  C194      kész, lemezstátusz előkészítése

****************************
EDB3   C8         INY
EDB4   B1 94      LDA  (94),Y    sáv és
EDB6   48         PHA
EDB7   C8         INY
EDB8   B1 94      LDA  (94),Y    a szektor tárolása
EDBA   48         PHA
EDBB   A0 13      LDY  #13       mutató az oldal-szektor-blokkon
EDBD   B1 94      LDA  (94),Y
EDBF   F0 0A      BEQ  EDCB      folytató-blokk?
EDC1   85 80      STA  80        sáv
EDC3   C8         INY
EDC4   B1 94      LDA  (94),Y
EDC6   85 81      STA  81        és az első oldal-szektor-blokk szektora
EDC8   20 E5 ED   JSR  EDE5      az oldal-szektor-blokkok lefoglalása
EDCB   68         PLA
EDCC   85 81      STA  81
EDCE   68         PLA            sáv és szektor visszaolvasása
EDCF   85 80      STA  80
EDD1   20 E5 ED   JSR  EDE5      a file-blokkok lefoglalása
EDD4   20 04 C6   JSR  C604      tart.jegyzékben az első bejegyzés elolvasása
EDD7   F0 C3      BEQ  ED9C      tartalomjegyzék vége?
EDD9   A0 00      LDY  #00
EDDB   B1 94      LDA  (94),Y    file-tipus
EDDD   30 D4      BMI  EDB3      7. bit magas, file lezárva?
EDDF   20 B6 C8   JSR  C8B6      file-tipus nulla, BAM felírása
EDE2   4C D4 ED   JMP  EDD4

****************************     az adatblokkok lefoglalása a BAM-ban
EDE5   20 5F D5   JSR  D55F      a sáv- és szektor-szám ellenőrzése
EDE8   20 90 EF   JSR  EF90      a blokk lefoglalása a BAM-ban
EDEB   20 75 D4   JSR  D475      a következő blokk olvasása
EDEE   A9 00      LDA  #00
EDF0   20 C8 D4   JSR  D4C8      puffer mutató nullán
EDF3   20 37 D1   JSR  D137      egy byte a pufferből
EDF6   85 80      STA  80        sáv
EDF8   20 37 D1   JSR  D137      egy byte a pufferből
```

```
EDFB  85 81     STA  81       szektor
EDFD  A5 80     LDA  80       további blokk következik?
EDFF  D0 03     BNE  EE04     igen
EE01  4C 27 D2  JMP  D227     csatornát lezárni!

EE04  20 90 EF  JSR  EF90     blokk lefoglalása a BAM-ban
EE07  20 4D D4  JSR  D44D     következő blokk olvasása
EE0A  4C EE ED  JMP  EDEE     tovább

******************************  N-utasítás, "Header"
EE0D  20 12 C3  JSR  C312     az egységszám beolvasása
EE10  A5 E2     LDA  E2
EE12  10 05     BPL  EE19     nem egyértelmű?
EE14  A9 33     LDA  #33
EE16  4C C8 C1  JMP  C1C8     33, "syntax error"

EE19  29 01     AND  #01
EE1B  85 7F     STA  7F       egységszám
EE1D  20 00 C1  JSR  C100     LED bekapcsolása
EE20  A5 7F     LDA  7F       az egységszám
EE22  0A        ASL  A        2-szer
EE23  AA        TAX
EE24  AC 7B 02  LDY  027B     vessző-pozició
EE27  CC 74 02  CPY  0274     a név végével összehasonlítani!
EE2A  F0 1A     BEQ  EE46     formátumszervezés ID nélkül
EE2C  B9 00 02  LDA  0200,Y   ID első karakterének
EE2F  95 12     STA  12,X     tárolása
EE31  B9 01 02  LDA  0201,Y   második karakter
EE34  95 13     STA  13,X
EE36  20 07 D3  JSR  D307     összes csatorna lezárása
EE39  A9 01     LDA  #01
EE3B  85 80     STA  80       1. sáv
EE3D  20 C6 C8  JSR  C8C6     a lemez formálása
EE40  20 05 F0  JSR  F005     a puffer törlése
EE43  4C 56 EE  JMP  EE56     tovább az alábbiak szerint

EE46  20 42 D0  JSR  D042     a BAM betöltése
EE49  A6 7F     LDX  7F       egységszám
EE4B  BD 01 01  LDA  0101,X
EE4E  CD D5 FE  CMP  FED5     "A", 1541-formátum jelölése
EE51  F0 03     BEQ  EE56     ok
EE53  4C 72 D5  JMP  D572     73, "cbm dos v2,6 1541"

EE56  20 B7 EE  JSR  EEB7     a BAM előállítása
EE59  A5 F9     LDA  F9       puffer-szám
EE5B  A8        TAY
EE5C  0A        ASL  A
EE5D  AA        TAX
EE5E  AD 88 FE  LDA  FE88     $90, a lemez nevének kezdete
EE61  95 99     STA  99,X     puffer mutató a néven
EE63  AE 7A 02  LDX  027A
EE66  A9 1B     LDA  #1B      27
EE68  20 6E C6  JSR  C66E     file-név beírása a pufferbe
EE6B  A0 12     LDY  #12      18. pozició
EE6D  A6 7F     LDX  7F       egységszám
EE6F  AD D5 FE  LDA  FED5     "A", 1541-formátum
EE72  9D 01 01  STA  0101,X
EE75  8A        TXA
EE76  0A        ASL  A        2-szer
EE77  AA        TAX
EE78  B5 12     LDA  12,X     ID, első karakter
EE7A  91 94     STA  (94),Y   a pufferben
EE7C  C8        INY
EE7D  B5 13     LDA  13,X     a második karakter
EE7F  91 94     STA  (94),Y   a pufferben
EE81  C8        INY
EE82  C8        INY
```

```
EE83  A9 32     LDA  #32      "2"
EE85  91 94     STA  (94),Y   a pufferben
EE87  C8        INY
EE88  AD D5 FE  LDA  FED5     "A", 1541-formátum
EE8B  91 94     STA  (94),Y   a pufferben
EE8D  A0 02     LDY  #02
EE8F  91 6D     STA  (6D),Y   és a 2. pozición
EE91  AD 85 FE  LDA  FE85     18
EE94  85 80     STA  80       sáv-szám
EE96  20 93 EF  JSR  EF93     a blokk lefoglalása
EE99  A9 01     LDA  #01      1
EE9B  85 81     STA  81       szektor-szám
EE9D  20 93 EF  JSR  EF93     blokk lefoglalása
EEA0  20 FF EE  JSR  EEFF     BAM felírása
EEA3  20 05 F0  JSR  F005     $6D/$6E mutató a pufferen, puffer törlése
EEA6  A0 01     LDY  #01
EEA8  A9 FF     LDA  #FF
EEAA  91 6D     STA  (6D),Y   Ø folytatósáv, $FF az érvényes byte-ok száma
EEAC  20 64 D4  JSR  D464     blokk felírása
EEAF  C6 81     DEC  81       szektor-szám csökkentése, Ø
EEB1  20 60 D4  JSR  D460     blokk olvasása
EEB4  4C 94 C1  JMP  C194     a lemezstátusz előkészitése

******************************  a BAM előállítása
EEB7  20 D1 F0  JSR  F0D1
EEBA  A0 00     LDY  #00
EEBC  A9 12     LDA  #12      18
EEBE  91 6D     STA  (6D),Y   mutató a tartalomjegyzék-sávra
EEC0  C8        INY
EEC1  98        TYA           1
EEC2  91 6D     STA  (6D),Y   mutató a tartalomjegyzék-szektorra
EEC4  C8        INY
EEC5  C8        INY
EEC6  C8        INY
EEC7  A9 00     LDA  #00
EEC9  85 6F     STA  6F
EECB  85 70     STA  70       3. byte a szektorok 24 bitjével azonos
EECD  85 71     STA  71
EECF  98        TYA           byte-pozició
EED0  4A        LSR  A
EED1  4A        LSR  A        sáv-szám: 4
EED2  20 4B F2  JSR  F24B     szektorok számának betöltése
EED5  91 6D     STA  (6D),Y   és beírása a BAM-ba
EED7  C8        INY
EED8  AA        TAX
EED9  38        SEC
EEDA  26 6F     ROL  6F
EEDC  26 70     ROL  70       a bitminta előállítása
EEDE  26 71     ROL  71
EEE0  CA        DEX
EEE1  D0 F6     BNE  EED9
EEE3  B5 6F     LDA  6F,X     3 byte
EEE5  91 6D     STA  (6D),Y   BAM a pufferben
EEE7  C8        INY
EEE8  E8        INX
EEE9  E0 03     CPX  #03
EEEB  90 F6     BCC  EEE3
EEED  C0 90     CPY  #90      már a 144-es pozició?
EEEF  90 D6     BCC  EEC7     nem, következő sáv
EEF1  4C 75 D0  JMP  D075     a szabad blokkok számának kiszámítása

******************************  szükség esetén BAM felülírása
EEF4  20 93 DF  JSR  DF93     a puffer-szám beolvasása
EEF7  AA        TAX
EEF8  BD 5B 02  LDA  025B,X   utasítás a lemezvezérlőnek!
EEFB  29 01     AND  #01
EEFD  85 7F     STA  7F       az egységszám leválasztása
```

```
EEFF  A4 7F     LDY  7F
EF01  B9 51 02  LDA  0251,Y     a BAM változtatási kapcsoló magas?
EF04  D0 01     BNE  EF07       igen
EF06  60        RTS

EF07  A9 00     LDA  #00
EF09  99 51 02  STA  0251,Y     a kapcsoló visszaállítása
EF0C  20 3A EF  JSR  EF3A       a BAM puffer mutatójának beállítása
EF0F  A5 7F     LDA  7F         egységszám
EF11  0A        ASL  A          2-szer
EF12  48        PHA
EF13  20 A5 F0  JSR  F0A5       BAM-bejegyzés ellenőrzése
EF16  68        PLA
EF17  18        CLC
EF18  69 01     ADC  #01        a sáv-szám növelése
EF1A  20 A5 F0  JSR  F0A5       a BAM-bejegyzés ellenőrzése
EF1D  A5 80     LDA  80         sáv
EF1F  48        PHA
EF20  A9 01     LDA  #01
EF22  85 80     STA  80         1. sáv
EF24  0A        ASL  A
EF25  0A        ASL  A          4-szer
EF26  85 6D     STA  6D
EF28  20 20 F2  JSR  F220       a BAM ellenőrzése
EF2B  E6 80     INC  80         sáv-szám növelése
EF2D  A5 80     LDA  80
EF2F  CD D7 FE  CMP  FED7       és a max. értékkel + 1 összehasonlítás 36-tal
EF32  90 F0     BCC  EF24       ok, következő sáv
EF34  68        PLA
EF35  85 80     STA  80         a sáv-szám visszaolvasása
EF37  4C 8A D5  JMP  D58A       BAM felírása a lemezre

******************************  a BAM puffer-mutatójának beállítása
EF3A  20 0F F1  JSR  F10F
EF3D  AA        TAX
EF3E  20 DF F0  JSR  F0DF       a puffer lefoglalása
EF41  A6 F9     LDX  F9         puffer-szám
EF43  BD E0 FE  LDA  FEE0,X     puffer-cím /Hi/
EF46  85 6E     STA  6E
EF48  A9 00     LDA  #00        /Lo/
EF4A  85 6D     STA  6D         mutató a $6D/$6E után
EF4C  60        RTS

******************************  tart.j-hez a szabad blokkok számának betöltése
EF4D  A6 7F     LDX  7F         egységszám
EF4F  BD FA 02  LDA  02FA,X     blokkok száma /Lo/
EF52  8D 72 02  STA  0272
EF55  BD FC 02  LDA  02FC,X     blokkok száma /Hi/
EF58  8D 73 02  STA  0273       a tartalomjegyzék a pufferben
EF5B  60        RTS

******************************  a blokkok szabad blokkok!
EF5C  20 F1 EF  JSR  EFF1       a puffer mutató beállítása
EF5F  20 CF EF  JSR  EFCF       a szektor bitjét a BAM-ban törölni!
EF62  38        SEC
EF63  D0 22     BNE  EF87       ha a blokk már szabad, akkor kész
EF65  B1 6D     LDA  (6D),Y     BAM bitmintája
EF67  1D E9 EF  ORA  EFE9,X     az X-bit beállítása
EF6A  91 6D     STA  (6D),Y
EF6C  20 88 EF  JSR  EF88       BAM-kapcsoló beállítása /módosítás volt!/
EF6F  A4 6F     LDY  6F
EF71  18        CLC
EF72  B1 6D     LDA  (6D),Y
EF74  69 01     ADC  #01        sávonkénti szabad blokkok számának növelése
EF76  91 6D     STA  (6D),Y
EF78  A5 80     LDA  80         sáv
EF7A  CD 85 FE  CMP  FE85       egyenlő 18-cal?
```

```
EF7D  F0 3B       BEQ  EFBA      ha igen, átmenni
EF7F  FE FA 02    INC  02FA,X    a lemez szabad blokkjainak számát növelni!
EF82  D0 03       BNE  EF87
EF84  FE FC 02    INC  02FC,X    blokkok számának növelése /Hi/
EF87  60          RTS

****************************     a BAM-kapcsoló beállítása /módositás történt/
EF88  A6 7F       LDX  7F        egységszám
EF8A  A9 01       LDA  #01
EF8C  9D 51 02    STA  0251,X    a kapcsoló = 1
EF8F  60          RTS

****************************     a blokk lefoglalása
EF90  20 F1 EF    JSR  EFF1      a puffer mutató beállítása
EF93  20 CF EF    JSR  EFCF      a szektor-bit törlése a BAM-ban
EF96  F0 36       BEQ  EFCE      ha már foglalt, kész
EF98  B1 6D       LDA  (6D),Y
EF9A  5D E9 EF    EOR  EFE9,X    a blokk-bit törlése
EF9D  91 6D       STA  (6D),Y
EF9F  20 88 EF    JSR  EF88      BAM-kapcsoló beállítása
EFA2  A4 6F       LDY  6F
EFA4  B1 6D       LDA  (6D),Y
EFA6  38          SEC
EFA7  E9 01       SBC  #01       sávonkénti blokkok számának csökkentése
EFA9  91 6D       STA  (6D),Y
EFAB  A5 80       LDA  80        sáv
EFAD  CD 85 FE    CMP  FE85      18?
EFB0  F0 0B       BEQ  EFBD      tartalomjegyzék-sávot kihagyni!
EFB2  BD FA 02    LDA  02FA,X    szabad blokkok száma /Lo/
EFB5  D0 03       BNE  EFBA
EFB7  DE FC 02    DEC  02FC,X    szabad blokkok számának csökkentése
EFBA  DE FA 02    DEC  02FA,X
EFBD  BD FC 02    LDA  02FC,X    szabad blokkok száma /Hi/
EFC0  D0 0C       BNE  EFCE      több mint 255 szabad blokk?
EFC2  BD FA 02    LDA  02FA,X    szabad blokkok száma /Lo/
EFC5  C9 03       CMP  #03
EFC7  B0 05       BCS  EFCE      3-nál kisebb?
EFC9  A9 72       LDA  #72
EFCB  20 C7 E6    JSR  E6C7      72, "disk full"
EFCE  60          RTS

****************************     a szektor bit törlése a sáv BAM mezőjének
EFCF  20 11 F0    JSR  F011      megkeresése
EFD2  98          TYA
EFD3  85 6F       STA  6F
EFD5  A5 81       LDA  81        a szektort
EFD7  4A          LSR  A
EFD8  4A          LSR  A         8-cal osztani!
EFD9  4A          LSR  A
EFDA  38          SEC
EFDB  65 6F       ADC  6F
EFDD  A8          TAY            byte-szám a BAM-bejegyzésben
EFDE  A5 81       LDA  81        szektor-szám
EFE0  29 07       AND  #07
EFE2  AA          TAX            bit-szám a BAM-bejegyzésben
EFE3  B1 6D       LDA  (6D),Y    byte a BAM-ban
EFE5  3D E9 EF    AND  EFE9,X    a szektor-bit törlése /foglalt!/
EFE8  60          RTS

****************************     második hatványok /négyzetek/
EFE9  01 02 04 08 10 20 40 80

****************************     változtatás után a BAM felírása
EFF1  A9 FF       LDA  #FF
EFF3  2C F9 02    BIT  02F9
EFF6  F0 0C       BEQ  F004
EFF8  10 0A       BPL  F004
```

```
EFFA  70 08      BVS   F004
EFFC  A9 00      LDA   #00
EFFE  8D F9 02   STA   02F9       a kapcsoló visszaállítása
F001  4C 8A D5   JMP   D58A       a blokk felírása

F004  60         RTS

*****************************     a BAM-puffer törlése
F005  20 3A EF   JSR   EF3A       $6D/$6E mutató a BAM-pufferen
F008  A0 00      LDY   #00
F00A  98         TYA
F00B  91 6D      STA   (6D),Y     BAM-puffer törlése
F00D  C8         INY
F00E  D0 FB      BNE   F00B
F010  60         RTS

*****************************
F011  A5 6F      LDA   6F
F013  48         PHA
F014  A5 70      LDA   70
F016  48         PHA
F017  A6 7F      LDX   7F         egységszám
F019  B5 FF      LDA   FF,X
F01B  F0 05      BEQ   F022       nulla?
F01D  A9 74      LDA   #74
F01F  20 48 E6   JSR   E648       "drive not ready"
F022  20 0F F1   JSR   F10F       a BAM-puffer számának beolvasása
F025  85 6F      STA   6F
F027  8A         TXA
F028  0A         ASL   A
F029  85 70      STA   70
F02B  AA         TAX
F02C  A5 80      LDA   80         sáv
F02E  DD 9D 02   CMP   029D,X
F031  F0 0B      BEQ   F03E
F033  E8         INX
F034  86 70      STX   70
F036  DD 9D 02   CMP   029D,X
F039  F0 03      BEQ   F03E
F03B  20 5B F0   JSR   F05B
F03E  A5 70      LDA   70
F040  A6 7F      LDX   7F         egységszám
F042  9D 9B 02   STA   029B,X
F045  0A         ASL   A
F046  0A         ASL   A          4-szer
F047  18         CLC
F048  69 A1      ADC   #A1
F04A  85 6D      STA   6D
F04C  A9 02      LDA   #02
F04E  69 00      ADC   #00
F050  85 6E      STA   6E
F052  A0 00      LDY   #00
F054  68         PLA
F055  85 70      STA   70
F057  68         PLA
F058  85 6F      STA   6F
F05A  60         RTS

*****************************
F05B  A6 6F      LDX   6F
F05D  20 DF F0   JSR   F0DF
F060  A5 7F      LDA   7F         egységszám
F062  AA         TAX
F063  0A         ASL   A
F064  1D 9B 02   ORA   029B,X
F067  49 01      EOR   #01
F069  29 03      AND   #03
```

```
F06B  85 70      STA  70
F06D  20 A5 F0   JSR  F0A5
F070  A5 F9      LDA  F9        puffer-szám
F072  0A         ASL  A
F073  AA         TAX
F074  A5 80      LDA  80        sáv
F076  0A         ASL  A
F077  0A         ASL  A         4-szer
F078  95 99      STA  99,X      a BAM-mezőn álló mutatóval azonos
F07A  A5 70      LDA  70
F07C  0A         ASL  A
F07D  0A         ASL  A
F07E  A8         TAY
F07F  A1 99      LDA  (99,X)
F081  99 A1 02   STA  02A1,Y
F084  A9 00      LDA  #00
F086  81 99      STA  (99,X)    nulla a pufferben
F088  F6 99      INC  99,X      a puffer mutató növelése
F08A  C8         INY
F08B  98         TYA
F08C  29 03      AND  #03
F08E  D0 EF      BNE  F07F
F090  A6 70      LDX  70
F092  A5 80      LDA  80        sáv
F094  9D 9D 02   STA  029D,X
F097  AD F9 02   LDA  02F9
F09A  D0 03      BNE  F09F
F09C  4C 8A D5   JMP  D58A      blokk felírása

F09F  09 80      ORA  #80
F0A1  8D F9 02   STA  02F9
F0A4  60         RTS

F0A5  A8         TAY
F0A6  B9 9D 02   LDA  029D,Y
F0A9  F0 25      BEQ  F0D0
F0AB  48         PHA
F0AC  A9 00      LDA  #00
F0AE  99 9D 02   STA  029D,Y
F0B1  A5 F9      LDA  F9        puffer-szám
F0B3  0A         ASL  A         2-szer
F0B4  AA         TAX
F0B5  68         PLA
F0B6  0A         ASL  A
F0B7  0A         ASL  A
F0B8  95 99      STA  99,X
F0BA  98         TYA
F0BB  0A         ASL  A
F0BC  0A         ASL  A
F0BD  A8         TAY
F0BE  B9 A1 02   LDA  02A1,Y
F0C1  81 99      STA  (99,X)    írás a pufferbe
F0C3  A9 00      LDA  #00
F0C5  99 A1 02   STA  02A1,Y
F0C8  F6 99      INC  99,X      a puffer mutató növelése
F0CA  C8         INY
F0CB  98         TYA
F0CC  29 03      AND  #03
F0CE  D0 EE      BNE  F0BE
F0D0  60         RTS

F0D1  A5 7F      LDA  7F        egységszám
F0D3  0A         ASL  A
F0D4  AA         TAX
F0D5  A9 00      LDA  #00
F0D7  9D 9D 02   STA  029D,X
F0DA  E8         INX
```

```
F0DB  9D 9D 02    STA   029D,X
F0DE  60          RTS

F0DF  B5 A7       LDA   A7,X
F0E1  C9 FF       CMP   #FF
F0E3  D0 25       BNE   F10A
F0E5  8A          TXA
F0E6  48          PHA
F0E7  20 8E D2    JSR   D28E
F0EA  AA          TAX
F0EB  10 05       BPL   F0F2
F0ED  A9 70       LDA   #70
F0EF  20 C8 C1    JSR   C1C8        70, "no channel"
F0F2  86 F9       STX   F9
F0F4  68          PLA
F0F5  A8          TAY
F0F6  8A          TXA
F0F7  09 80       ORA   #80
F0F9  99 A7 00    STA   00A7,Y
F0FC  0A          ASL   A
F0FD  AA          TAX
F0FE  AD 85 FE    LDA   FE85        18, tartalomjegyzék-sáv
F101  95 06       STA   06,X        tárolása
F103  A9 00       LDA   #00         0
F105  95 07       STA   07,X        szektorként
F107  4C 86 D5    JMP   D586        blokk felírása

F10A  29 0F       AND   #0F
F10C  85 F9       STA   F9          puffer-szám
F10E  60          RTS

******************************      a BAM puffer-számának beolvasása
F10F  A9 06       LDA   #06
F111  A6 7F       LDX   7F          egységszám
F113  D0 03       BNE   F118
F115  18          CLC
F116  69 07       ADC   #07         0. meghajtóra 13-at ad
F118  60          RTS

******************************      a BAM puffer-száma X után
F119  20 0F F1    JSR   F10F        a puffer-szám betöltése
F11C  AA          TAX
F11D  60          RTS

******************************      szabad blokk keresése a BAM-ban, lefoglalása
F11E  20 3E DE    JSR   DE3E        a sáv- és a szektor-szám beolvasása
F121  A9 03       LDA   #03
F123  85 6F       STA   6F          számláló
F125  A9 01       LDA   #01
F127  0D F9 02    ORA   02F9
F12A  8D F9 02    STA   02F9
F12D  A5 6F       LDA   6F          a számláló tárolása
F12F  48          PHA
F130  20 11 F0    JSR   F011        ehhez a sávhoz megkeresni a BAM-mezőt!
F133  68          PLA
F134  85 6F       STA   6F          a számláló visszaolvasása
F136  B1 6D       LDA   (6D),Y      sáv szabad byte-jainak száma
F138  D0 39       BNE   F173        vannak még szabad blokkok?
F13A  A5 80       LDA   80          sáv
F13C  CD 85 FE    CMP   FE85        18, tartalomjegyzék-sáv
F13F  F0 19       BEQ   F15A        igen, "disk full"
F141  90 1C       BCC   F15F        ha kisebb, akkor a következő alacsonyabb sáv
F143  E6 80       INC   80          sáv-szám növelése
F145  A5 80       LDA   80
F147  CD D7 FE    CMP   FED7        36, legmagasabb sáv-szám plusz egy
F14A  D0 E1       BNE   F12D        nem, ezen a sávon tovább keresni
F14C  AE 85 FE    LDX   FE85        18, tartalomjegyzék-sáv
```

```
F14F  CA          DEX              a csökkentett értéket
F150  86 80       STX   80         sáv-számként tárolni!
F152  A9 00       LDA   #00
F154  85 81       STA   81         a szektor-szám kezdőértéke nulla
F156  C6 6F       DEC   6F         számlálót csökkenteni, ha
F158  D0 D3       BNE   F12D       még mindig nulla, tovább keresni!
F15A  A9 72       LDA   #72
F15C  20 C8 C1    JSR   C1C8       62, "disk full"
F15F  C6 80       DEC   80         sáv-szám csökkentése
F161  D0 CA       BNE   F12D       ha még mindig nulla, ezen a sávon keresni!
F163  AE 85 FE    LDX   FE85       18, tartalomjegyzék sáv
F166  E8          INX              a növelt értéket
F167  86 80       STX   80         sáv-számként tárolni
F169  A9 00       LDA   #00
F16B  85 81       STA   81         szektor kezdőértéke: nulla
F16D  C6 6F       DEC   6F         számláló csökkentése
F16F  D0 BC       BNE   F12D       ha még mindig nulla, tovább keresni!
F171  F0 E7       BEQ   F15A       egyébként "disk full"
F173  A5 81       LDA   81         szektor-szám
F175  18          CLC
F176  65 69       ADC   69         plusz /10/ lépéstávolság
F178  85 81       STA   81         mint új szám
F17A  A5 80       LDA   80         sáv-szám
F17C  20 4B F2    JSR   F24B       behozni a maximális szektor-számot!
F17F  8D 4E 02    STA   024E
F182  8D 4D 02    STA   024D       és tárolni!
F185  C5 81       CMP   81         nagyobb mint a megválasztott szektor-szám?
F187  B0 0C       BCS   F195       igen
F189  38          SEC              egyébként
F18A  A5 81       LDA   81         szektor-szám
F18C  ED 4E 02    SBC   024E       minusz a maximális szektor-szám
F18F  85 81       STA   81         Uj szektor-számként tárolni!
F191  F0 02       BEQ   F195       nulla?
F193  C6 81       DEC   81         egyébként a szektor-számot 1-gyel csökkenteni!
F195  20 FA F1    JSR   F1FA       BAM-ot ellenőrizni, szabad szektort keresni!
F198  F0 03       BEQ   F19D       nincs meg?
F19A  4C 90 EF    JMP   EF90       BAM-ban lefoglalni a blokkot!

F19D  A9 00       LDA   #00
F19F  85 81       STA   81         nulla szektor
F1A1  20 FA F1    JSR   F1FA       szabad szektor keresése
F1A4  D0 F4       BNE   F19A       megvan?
F1A6  4C F5 F1    JMP   F1F5       nincs, "dir error"

****************************       szabad szektor keresése és lefoglalása
F1A9  A9 01       LDA   #01
F1AB  0D F9 02    ORA   02F9
F1AE  8D F9 02    STA   02F9
F1B1  A5 86       LDA   86
F1B3  48          PHA
F1B4  A9 01       LDA   #01        sáv-számláló
F1B6  85 86       STA   86
F1B8  AD 85 FE    LDA   FE85       18, tartalomjegyzék-sáv
F1BB  38          SEC
F1BC  E5 86       SBC   86         minusz számláló
F1BE  85 80       STA   80         sáv-számként tárolni!
F1C0  90 09       BCC   F1CB       az eredmény kisebb nullánál?
F1C2  F0 07       BEQ   F1CB       ha igen a tart.jegyz. fölött keresni!
F1C4  20 11 F0    JSR   F011       a BAM-mező megkeresése ehhez a sávhoz
F1C7  B1 6D       LDA   (6D),Y     szabad blokkok száma?
F1C9  D0 1B       BNE   F1E6       vannak szabad blokkok?
F1CB  AD 85 FE    LDA   FE85       18, tartalomjegyzék-sáv
F1CE  18          CLC
F1CF  65 86       ADC   86         plusz számláló
F1D1  85 80       STA   80         sáv-számként tárolni!
F1D3  E6 86       INC   86         számlálót növelni
F1D5  CD D7 FE    CMP   FED7       36, maximális sáv-szám plusz egy
```

```
F1D8  90 05     BCC  F1DF       ha kisebb, ok
F1DA  A9 67     LDA  #67
F1DC  20 45 E6  JSR  E645       67, "illegal track or sector" '
F1DF  20 11 F0  JSR  F011       BAM-mezőt keresni ehhez a sávhoz
F1E2  B1 6D     LDA  (6D),Y     szabad blokkok száma?
F1E4  F0 D2     BEQ  F1B8       nincs már szabad blokk?
F1E6  68        PLA
F1E7  85 86     STA  86
F1E9  A9 00     LDA  #00
F1EB  85 81     STA  81         Ø. szektor
F1ED  20 FA F1  JSR  F1FA       szabad blokkot keresni!
F1F0  F0 03     BEQ  F1F5       nincs
F1F2  4C 90 EF  JMP  EF90       lefoglalni a blokkot a BAM-ban!

F1F5  A9 71     LDA  #71
F1F7  20 45 E6  JSR  E645       71, "dir error"

******************************  szabad blokk keresése az aktuális sávon
F1FA  20 11 F0  JSR  F011       ehhez a sávhoz BAM-mezőt keresni
F1FD  98        TYA             a szabad blokk számát jelzi
F1FE  48        PHA
F1FF  20 20 F2  JSR  F220       BAM-ot ellenőrizni
F202  A5 80     LDA  80         sáv
F204  20 4B F2  JSR  F24B       behozni a sáv maximális szektor-számát
F207  8D 4E 02  STA  024E       tárolni!
F20A  68        PLA
F20B  85 6F     STA  6F         a mutató tárolása
F20D  A5 81     LDA  81         a szektor
F20F  CD 4E 02  CMP  024E       összehasonlítása a maximális számmal
F212  B0 09     BCS  F21D       nagyobb vagy egyenlő?
F214  20 D5 EF  JSR  EFD5       a szektor bit-számának beolvasása
F217  D0 06     BNE  F21F       szabad szektor?
F219  E6 81     INC  81         szektor-számot növelni!
F21B  D0 F0     BNE  F20D       és ellenőrizni, hogy szabad-e
F21D  A9 00     LDA  #00        nincs szabad szektor
F21F  60        RTS

******************************  szabd blokkok számának ellenőrzése a BAM-ban
F220  A5 6F     LDA  6F
F222  48        PHA
F223  A9 00     LDA  #00
F225  85 6F     STA  6F         számláló nullán
F227  AC 86 FE  LDY  FE86       4, sávonkénti byte-ok száma a BAM-ban
F22A  88        DEY
F22B  A2 07     LDX  #07
F22D  B1 6D     LDA  (6D),Y
F22F  3D E9 EF  AND  EFE9,X     a bit leválasztása
F232  F0 02     BEQ  F236
F234  E6 6F     INC  6F         szabad szektor esetén a számláló növelése
F236  CA        DEX
F237  10 F4     BPL  F22D
F239  88        DEY
F23A  D0 EF     BNE  F22B
F23C  B1 6D     LDA  (6D),Y     összehasonlítása a lemezen levő szektor-
F23E  C5 6F     CMP  6F         számmal
F240  D0 04     BNE  F246       ha, nem egyenlő, hiba!
F242  68        PLA
F243  85 6F     STA  6F
F245  60        RTS

F246  A9 71     LDA  #71
F248  20 45 E6  JSR  E645       71, "dir error"

******************************  sávonkénti szektorok számának meghatározása
F24B  AE D6 FE  LDX  FED6       4 különböző érték
F24E  DD D6 FE  CMP  FED6,X     sáv-szám
F251  CA        DEX
```

```
F252  B0 FA     BCS   F24E     még mindig nagyobb?
F254  BD D1 FE  LDA   FED1,X   a szektorok számának betöltése
F257  60        RTS

F258  60        RTS

******************************  inicializálás
F259  A9 6F     LDA   #6F      a 4. bit /Write Protect/ és 7. /SYNC/ bemenet
F25B  8D 02 1C  STA   1C02     B-kapu az adatirány-regiszter
F25E  29 F0     AND   #F0
F260  8D 00 1C  STA   1C00     B-kapu, vezérlőkapu
F263  AD 0C 1C  LDA   1C0C     PCR, ellenőrzőregiszter
F266  29 FE     AND   #FE
F268  09 0E     ORA   #0E
F26A  09 E0     ORA   #E0
F26C  8D 0C 1C  STA   1C0C
F26F  A9 41     LDA   #41
F271  8D 0B 1C  STA   1C0B
F274  A9 00     LDA   #00
F276  8D 06 1C  STA   1C06
F279  A9 3A     LDA   #3A
F27B  8D 07 1C  STA   1C07
F27E  8D 05 1C  STA   1C05
F281  A9 7F     LDA   #7F
F283  8D 0E 1C  STA   1C0E     az IRQ-k törlése
F286  A9 C0     LDA   #C0
F288  8D 0D 1C  STA   1C0D
F28B  8D 0E 1C  STA   1C0E     IER, megszakítások lehetségesek
F28E  A9 FF     LDA   #FF
F290  85 3E     STA   3E
F292  85 51     STA   51       sáv-számláló formáláshoz
F294  A9 08     LDA   #08      8
F296  85 39     STA   39       blokk-fejléc állandói
F298  A9 07     LDA   #07      7
F29A  85 47     STA   47       adatblokk állandói
F29C  A9 05     LDA   #05
F29E  85 62     STA   62
F2A0  A9 FA     LDA   #FA      $62/$63 mutató $FA05-ön
F2A2  85 63     STA   63
F2A4  A9 C8     LDA   #C8      200
F2A6  85 64     STA   64
F2A8  A9 04     LDA   #04
F2AA  85 5E     STA   5E
F2AC  A9 04     LDA   #04
F2AE  85 5F     STA   5F

******************************  IRQ-rutin a lemezvezérlőhöz
F2B0  BA        TSX
F2B1  86 49     STX   49       verem-mutató tárolása
F2B3  AD 04 1C  LDA   1C04
F2B6  AD 0C 1C  LDA   1C0C     timer megszakítás-kapcsoló törlése
F2B9  09 0E     ORA   #0E
F2BB  8D 0C 1C  STA   1C0C
F2BE  A0 05     LDY   #05
F2C0  B9 00 00  LDA   0000,Y   Y puffer feladata?
F2C3  10 2E     BPL   F2F3     nem
F2C5  C9 D0     CMP   #D0      a programkódot a pufferben kell kialakítani?
F2C7  D0 04     BNE   F2CD     nem
F2C9  98        TYA
F2CA  4C 70 F3  JMP   F370     programot a pufferben futtatni!

F2CD  29 01     AND   #01      az egységszám leválasztása
F2CF  F0 07     BEQ   F2D8     nullás egység?
F2D1  84 3F     STY   3F
F2D3  A9 0F     LDA   #0F      ha nem,
F2D5  4C 69 F9  JMP   F969     74, "drive not ready"
```

```
F2D8  AA        TAX
F2D9  85 3D     STA  3D
F2DB  C5 3E     CMP  3E         jár a motor?
F2DD  F0 0A     BEQ  F2E9       igen
F2DF  20 7E F9  JSR  F97E       a meghajtóegység motorjának bekapcsolása
F2E2  A5 3D     LDA  3D
F2E4  85 3E     STA  3E         a kapcsoló beállítása
F2E6  4C 9C F9  JMP  F99C       a job-ciklushoz

F2E9  A5 20     LDA  20
F2EB  30 03     BMI  F2F0       a fejpozicionálás programozva?
F2ED  0A        ASL  A
F2EE  10 09     BPL  F2F9
F2F0  4C 9C F9  JMP  F99C       job-ciklushoz

F2F3  88        DEY
F2F4  10 CA     BPL  F2C0       a következő puffer ellenőrzése
F2F6  4C 9C F9  JMP  F99C       job-ciklushoz

F2F9  A9 20     LDA  #20
F2FB  85 20     STA  20         fejpozicionálást programozni!
F2FD  A0 05     LDY  #05
F2FF  84 3F     STY  3F         a puffer-számláló inicializálása
F301  20 93 F3  JSR  F393       mutató beállítása
F304  30 1A     BMI  F320       van a puffer számára feladat?
F306  C6 3F     DEC  3F         számláló csökkentése
F308  10 F7     BPL  F301       következő puffer ellenőrzése
F30A  A4 41     LDY  41         puffer-szám
F30C  20 95 F3  JSR  F395       mutató beállítása a pufferre
F30F  A5 42     LDA  42
F311  85 4A     STA  4A         mint a fejpozicionálás számlálója
F313  06 4A     ASL  4A
F315  A9 60     LDA  #60        fejpozicionálás kapcsolójának beállítása
F317  85 20     STA  20
F319  B1 32     LDA  (32),Y     sáv-szám beolvasása a pufferból
F31B  85 22     STA  22
F31D  4C 9C F9  JMP  F99C       job-ciklushoz

F320  29 01     AND  #01        egységszám leválasztása
F322  C5 3D     CMP  3D         egyenlő az utolsó job meghajtóegység számával?
F324  D0 E0     BNE  F306       nem
F326  A5 22     LDA  22         utolsó sáv-szám
F328  F0 12     BEQ  F33C       egyenlő nullával?
F32A  38        SEC
F32B  F1 32     SBC  (32),Y     egyenlő a job sáv-számával?
F32D  F0 0D     BEQ  F33C       igen
F32F  49 FF     EOR  #FF
F331  85 42     STA  42
F333  E6 42     INC  42
F335  A5 3F     LDA  3F         egységszám
F337  85 41     STA  41
F339  4C 06 F3  JMP  F306       további ellenőrzés

F33C  A2 04     LDX  #04
F33E  B1 32     LDA  (32),Y     job sáv-szám
F340  85 40     STA  40         tárolása
F342  DD D6 FE  CMP  FED6,X     összehasonlítása a maximális sáv-számmal
F345  CA        DEX
F346  B0 FA     BCS  F342       nagyobb?
F348  BD D1 FE  LDA  FED1,X     sávonkénti szektor-szám beolvasása
F34B  85 43     STA  43         és tárolása
F34D  8A        TXA
F34E  0A        ASL  A
F34F  0A        ASL  A
F350  0A        ASL  A          sáv-tartomány száma, 32-szer
F351  0A        ASL  A
F352  0A        ASL  A
```

```
F353  85 44      STA  44        Ø-t, 32-t, 64-t, 96-t ad
F355  AD ØØ 1C   LDA  1CØØ
F358  29 9F      AND  #9F
F35A  Ø5 44      ORA  44        a motor vezérlő-byte generálása
F35C  8D ØØ 1C   STA  1CØØ
F35F  A6 3D      LDX  3D
F361  A5 45      LDA  45        utasításkód
F363  C9 4Ø      CMP  #4Ø       mágnesfejet pozicionálni?
F365  FØ 15      BEQ  F37C      igen
F367  C9 6Ø      CMP  #6Ø       program utasításkódját a pufferben futtatni?
F369  FØ Ø3      BEQ  F36E      igen
F36B  4C B1 F3   JMP  F3B1      olvasni a blokk-fejlécet!

*****************************   program futtatása a pufferben
F36E  A5 3F      LDA  3F        puffer-szám
F370  18         CLC
F371  69 Ø3      ADC  #Ø3       plusz 3
F373  85 31      STA  31
F375  A9 ØØ      LDA  #ØØ       egyenlő a puffer címével
F377  85 3Ø      STA  3Ø
F379  6C 3Ø ØØ   JMP  (ØØ3Ø)    program futtatása a pufferben

*****************************   mágnesfej pozicionálása
F37C  A9 6Ø      LDA  #6Ø
F37E  85 2Ø      STA  2Ø        fejpozicionálás kapcsoló beállítása
F38Ø  AD ØØ 1C   LDA  1CØØ
F383  29 FC      AND  #FC       léptetőmotorok bekapcsolása
F385  8D ØØ 1C   STA  1CØØ
F388  A9 A4      LDA  #A4       164
F38A  85 4A      STA  4A        fejpozicionálás léptetés-számlálója
F38C  A9 Ø1      LDA  #Ø1
F38E  85 22      STA  22        sáv-szám
F39Ø  4C 69 F9   JMP  F969      ok

*****************************   a mutató inicializálása a pufferben
F393  A4 3F      LDY  3F        puffer-szám
F395  B9 ØØ ØØ   LDA  ØØØØ,Y    utasításkód
F398  48         PHA            tárolása
F399  1Ø 1Ø      BPL  F3AB
F39B  29 78      AND  #78       Ø, 1, 2. és 7. bit törlése
F39D  85 45      STA  45
F39F  98         TYA            puffer-szám
F3AØ  ØA         ASL  A         2-szer
F3A1  69 Ø6      ADC  #Ø6       plusz 6
F3A3  85 32      STA  32        egyenlő az aktuális pufferen álló mutatóval
F3A5  98         TYA            puffer-szám
F3A6  18         CLC
F3A7  69 Ø3      ADC  #Ø3       plusz 3
F3A9  85 31      STA  31        egyenlő a puffer-címmel /Hi/
F3AB  AØ ØØ      LDY  #ØØ
F3AD  84 3Ø      STY  3Ø        puffer-cím /Lo/
F3AF  68         PLA            utasításkód visszaolvasása
F3BØ  6Ø         RTS

*****************************   blokk-fejléc olvasása, ID ellenőrzése
F3B1  A2 5A      LDX  #5A       9Ø
F3B3  86 4B      STX  4B        számláló
F3B5  A2 ØØ      LDX  #ØØ
F3B7  A9 52      LDA  #52       82
F3B9  85 24      STA  24
F3BB  2Ø 56 F5   JSR  F556      megvárni az SYNC-t
F3BE  5Ø FE      BVC  F3BE      Byte Ready?
F3CØ  B8         CLV
F3C1  AD Ø1 1C   LDA  1CØ1      adatok az olvasófejtől
F3C4  C5 24      CMP  24
F3C6  DØ 3F      BNE  F4Ø7      2Ø, "read error"
F3C8  5Ø FE      BVC  F3C8      Byte Ready?
```

```
F3CA  B8          CLV
F3CB  AD 01 1C    LDA   1C01      adatbyte a lemezről /blokk-fejléc/
F3CE  95 25       STA   25,X      7 byte tárolása
F3D0  E8          INX
F3D1  E0 07       CPX   #07
F3D3  D0 F3       BNE   F3C8      a beolvasás folytatása
F3D5  20 97 F4    JSR   F497
F3D8  A0 04       LDY   #04       4 byte plusz paritás
F3DA  A9 00       LDA   #00
F3DC  59 16 00    EOR   0016,Y    a fejléc ellenőrző összegének kialakítása
F3DF  88          DEY
F3E0  10 FA       BPL   F3DC
F3E2  C9 00       CMP   #00       rendben van a paritás?
F3E4  D0 38       BNE   F41E      27, "write error"
F3E6  A6 3E       LDX   3E        egységszám
F3E8  A5 18       LDA   18        fejléc sáv-száma
F3EA  95 22       STA   22,X      aktuális sáv-számként átvenni!
F3EC  A5 45       LDA   45
F3EE  C9 30       CMP   #30       "fejlécet átvenni" kód
F3F0  F0 1E       BEQ   F410      fejlécet átvenni!
F3F2  A5 3E       LDA   3E
F3F4  0A          ASL   A
F3F5  A8          TAY
F3F6  B9 12 00    LDA   0012,Y
F3F9  C5 16       CMP   16        ID1 ellenőrzése
F3FB  D0 1E       BNE   F41B
F3FD  B9 13 00    LDA   0013,Y
F400  C5 17       CMP   17        ID2 ellenőrzése
F402  D0 17       BNE   F41B      ha nem egyenlő, akkor 29,"disk id mismatch"
F404  4C 23 F4    JMP   F423

F407  C6 4B       DEC   4B        keresés-számláló csökkentése
F409  D0 B0       BNE   F3BB      és mégegyszer megpróbálni!
F40B  A9 02       LDA   #02       egyébként
F40D  20 69 F9    JSR   F969      20, "read error"

*****************************     a blokk fejlécének beolvasása
F410  A5 16       LDA   16        ID1
F412  85 12       STA   12
F414  A5 17       LDA   17        és ID2-t
F416  85 13       STA   13        átvenni!
F418  A9 01       LDA   #01       ok
F41A  2C          .BYTE $2C
F41B  A9 0B       LDA   #0B       29, "disk id mismatch"
F41D  2C          .BYTE $2C
F41E  A9 09       LDA   #09       27, "write error"
F41D  2C A9 09    BIT   09A9
F420  4C 69 F9    JMP   F969      lezárás

*****************************
F423  A9 7F       LDA   #7F
F425  85 4C       STA   4C
F427  A5 19       LDA   19
F429  18          CLC
F42A  69 02       ADC   #02
F42C  C5 43       CMP   43
F42E  90 02       BCC   F432
F430  E5 43       SBC   43
F432  85 4D       STA   4D
F434  A2 05       LDX   #05
F436  86 3F       STX   3F
F438  A2 FF       LDX   #FF
F43A  20 93 F3    JSR   F393      a lemezvezérlő puffermutatójának beállítása
F43D  10 44       BPL   F483
F43F  85 44       STA   44
F441  29 01       AND   #01
F443  C5 3E       CMP   3E
```

```
F445  D0 3C     BNE  F483
F447  A0 00     LDY  #00
F449  B1 32     LDA  (32),Y
F44B  C5 40     CMP  40
F44D  D0 34     BNE  F483
F44F  A5 45     LDA  45      utasításkód
F451  C9 60     CMP  #60
F453  F0 0C     BEQ  F461
F455  A0 01     LDY  #01
F457  38        SEC
F458  B1 32     LDA  (32),Y
F45A  E5 4D     SBC  4D
F45C  10 03     BPL  F461
F45E  18        CLC
F45F  65 43     ADC  43
F461  C5 4C     CMP  4C
F463  B0 1E     BCS  F483
F465  48        PHA
F466  A5 45     LDA  45
F468  F0 14     BEQ  F47E
F46A  68        PLA
F46B  C9 09     CMP  #09
F46D  90 14     BCC  F483
F46F  C9 0C     CMP  #0C
F471  B0 10     BCS  F483
F473  85 4C     STA  4C
F475  A5 3F     LDA  3F
F477  AA        TAX
F478  69 03     ADC  #03
F47A  85 31     STA  31
F47C  D0 05     BNE  F483
F47E  68        PLA
F47F  C9 06     CMP  #06
F481  90 F0     BCC  F473
F483  C6 3F     DEC  3F
F485  10 B3     BPL  F43A
F487  8A        TXA
F488  10 03     BPL  F48D
F48A  4C 9C F9  JMP  F99C    job-ciklushoz

F48D  86 3F     STX  3F
F48F  20 93 F3  JSR  F393    a puffer-szám beolvasása
F492  A5 45     LDA  45      utasításkód
F494  4C CA F4  JMP  F4CA    az ellenőrzés folytatása

F497  A5 30     LDA  30
F499  48        PHA          a $30/$31 mutató tárolása
F49A  A5 31     LDA  31
F49C  48        PHA
F49D  A9 24     LDA  #24
F49F  85 30     STA  30
F4A1  A9 00     LDA  #00     $30/$31 mutató $24-en
F4A3  85 31     STA  31
F4A5  A9 00     LDA  #00
F4A7  85 34     STA  34
F4A9  20 E6 F7  JSR  F7E6
F4AC  A5 55     LDA  55
F4AE  85 18     STA  18
F4B0  A5 54     LDA  54
F4B2  85 19     STA  19
F4B4  A5 53     LDA  53
F4B6  85 1A     STA  1A
F4B8  20 E6 F7  JSR  F7E6
F4BB  A5 52     LDA  52
F4BD  85 17     STA  17
F4BF  A5 53     LDA  53
F4C1  85 16     STA  16
```

```
F4C3  68          PLA
F4C4  85 31       STA    31
F4C6  68          PLA             a $30/$31 mutató visszatöltése
F4C7  85 30       STA    30
F4C9  60          RTS

****************************
F4CA  C9 00       CMP   #00       "olvasás" utasításkód?
F4CC  F0 03       BEQ    F4D1     igen
F4CE  4C 6E F5    JMP    F56E     tovább ellenőrizni az utasításkódot!

F4D1  20 0A F5    JSR    F50A     megkeresni az adatblokk kezdetét!
F4D4  50 FE       BVC    F4D4     Byte Ready?
F4D6  B8          CLV
F4D7  AD 01 1C    LDA    1C01     adatbyte beolvasása
F4DA  91 30       STA   (30),Y    a pufferbe
F4DC  C8          INY             256-szor
F4DD  D0 F5       BNE    F4D4
F4DF  A0 BA       LDY   #BA
F4E1  50 FE       BVC    F4E1     Byte Ready?
F4E3  B8          CLV
F4E4  AD 01 1C    LDA    1C01     byte olvasása
F4E7  99 00 01    STA    0100,Y   $1BA-tól $1FF-ig
F4EA  C8          INY
F4EB  D0 F4       BNE    F4E1
F4ED  20 E0 F8    JSR    F8E0
F4F0  A5 38       LDA    38
F4F2  C5 47       CMP    47       7-tel egyenlő, adatblokk kezdete?
F4F4  F0 05       BEQ    F4FB     igen
F4F6  A9 04       LDA   #04       22, "read error"
F4F8  4C 69 F9    JMP    F969     hibalezárás

F4FB  20 E9 F5    JSR    F5E9     adatblokk paritásának kiszámitása
F4FE  C5 3A       CMP    3A       egyezik?
F500  F0 03       BEQ    F505     igen
F502  A9 05       LDA   #05       23, "read error"
F504  2C A9 01    BIT    01A9     ok
F507  4C 69 F9    JMP    F969     hibaüzenet előkészítése

****************************      az adatblokk kezdete
F50A  20 10 F5    JSR    F510     blokk-fejléc olvasás
F50D  4C 56 F5    JMP    F556     kivárni az SYNC-t

****************************      a blokk-fejléc olvasása
F510  A5 3D       LDA    3D       egységszám
F512  0A          ASL    A
F513  AA          TAX
F514  B5 12       LDA    12,X     ID1
F516  85 16       STA    16       tárolása
F518  B5 13       LDA    13,X     ID2
F51A  85 17       STA    17       tárolása
F51C  A0 00       LDY   #00
F51E  B1 32       LDA   (32),Y    sáv
F520  85 18       STA    18
F522  C8          INY
F523  B1 32       LDA   (32),Y    a szektor-szám beolvasása a pufferból
F525  85 19       STA    19
F527  A9 00       LDA   #00
F529  45 16       EOR    16
F52B  45 17       EOR    17       a blokk-fejléc paritásának kiszámítása
F52D  45 18       EOR    18
F52F  45 19       EOR    19
F531  85 1A       STA    1A       és tárolása
F533  20 34 F9    JSR    F934
F536  A2 5A       LDX   #5A       90 kisérlet
F538  20 56 F5    JSR    F556     kivárni az SYNC-t!
F53B  A0 00       LDY   #00
```

```
F53D  50 FE      BVC   F53D       Byte Ready?
F53F  B8         CLV
F540  AD 01 1C   LDA   1C01       adatok olvasása a blokk-fejlécből
F543  D9 24 00   CMP   0024,Y     összehasonlítás a tárolt adatokkal
F546  D0 06      BNE   F54E       ha nem egyenlő mégegyszer megkisérelni!
F548  C8         INY
F549  C0 08      CPY   #08        megtörtént 8 byte olvasása?
F54B  D0 F0      BNE   F53D       nem
F54D  60         RTS

F54E  CA         DEX              a számláló csökkentése
F54F  D0 E7      BNE   F538       még nem nulla?
F551  A9 02      LDA   #02
F553  4C 69 F9   JMP   F969       20, "read error"

******************************    megvárni az SYNC-t!
F556  A9 D0      LDA   #D0        2Ø8
F558  8D 05 18   STA   1805       az óra indítása
F55B  A9 03      LDA   #03        hibakód
F55D  2C 05 18   BIT   1805
F560  10 F1      BPL   F553       ha az óra lejárt, 21, "read error"
F562  2C 00 1C   BIT   1C00       SYNC-jel
F565  30 F6      BMI   F55D       még nincs meg?
F567  AD 01 1C   LDA   1C01       byte olvasása
F56A  B8         CLV
F56B  A0 00      LDY   #00
F56D  60         RTS

******************************
F56E  C9 10      CMP   #10        írás-utasításkód
F570  F0 03      BEQ   F575       igen
F572  4C 91 F6   JMP   F691       az utasításkód további ellenőrzése

******************************    adatblokk felírása a lemezre
F575  20 E9 F5   JSR   F5E9       a puffer paritásának kiszámítása
F578  85 3A      STA   3A         és tárolása
F57A  AD 00 1C   LDA   1C00       B-kapu olvasása
F57D  29 10      AND   #10        a "Write Protect" bit leválasztása
F57F  D0 05      BNE   F586       nincs beállítva, ok
F581  A9 08      LDA   #08
F583  4C 69 F9   JMP   F969       26, "write protect on"

F586  20 8F F7   JSR   F78F
F589  20 10 F5   JSR   F510       a blokk-fejléc megkeresése
F58C  A2 09      LDX   #09
F58E  50 FE      BVC   F58E       Byte Ready?
F590  B8         CLV
F591  CA         DEX              blokk-fejléc utáni 9 byte átolvasása
F592  D0 FA      BNE   F58E
F594  A9 FF      LDA   #FF
F596  8D 03 1C   STA   1C03       "A"-kaput /író-/olvasófej/ kimenetre
F599  AD 0C 1C   LDA   1C0C
F59C  29 1F      AND   #1F
F59E  09 C0      ORA   #C0        PCR-t outputra átkapcsolni!
F5A0  8D 0C 1C   STA   1C0C
F5A3  A9 FF      LDA   #FF
F5A5  A2 05      LDX   #05
F5A7  8D 01 1C   STA   1C01       $FF 5-szöri felírása a lemezre
F5AA  B8         CLV
F5AB  50 FE      BVC   F5AB       SYNC-jelként
F5AD  B8         CLV
F5AE  CA         DEX
F5AF  D0 FA      BNE   F5AB
F5B1  A0 BB      LDY   #BB
F5B3  B9 00 01   LDA   0100,Y     $1BB - $1FF byte-ok a lemezen
F5B6  50 FE      BVC   F5B6
F5B8  B8         CLV
```

```
F5B9   8D 01 1C    STA   1C01
F5BC   C8          INY
F5BD   D0 F4       BNE   F5B3
F5BF   B1 30       LDA   (30),Y      adatpuffer /256 byte/ felírása a lemezre
F5C1   50 FE       BVC   F5C1
F5C3   B8          CLV
F5C4   8D 01 1C    STA   1C01
F5C7   C8          INY
F5C8   D0 F5       BNE   F5BF
F5CA   50 FE       BVC   F5CA        Byte Ready?
F5CC   AD 0C 1C    LDA   1C0C
F5CF   09 E0       ORA   #E0         PCR ismét inputon
F5D1   8D 0C 1C    STA   1C0C
F5D4   A9 00       LDA   #00
F5D6   8D 03 1C    STA   1C03        "A"-kapu /író-/olvasófej/ bemeneten
F5D9   20 F2 F5    JSR   F5F2
F5DC   A4 3F       LDY   3F
F5DE   B9 00 00    LDA   0000,Y
F5E1   49 30       EOR   #30         írás-utasításkódot "Verify"-re átalakítani!
F5E3   99 00 00    STA   0000,Y
F5E6   4C B1 F3    JMP   F3B1

*****************************        adat-puffer paritásának kiszámítása
F5E9   A9 00       LDA   #00
F5EB   A8          TAY
F5EC   51 30       EOR   (30),Y
F5EE   C8          INY
F5EF   D0 FB       BNE   F5EC
F5F1   60          RTS

F5F2   A9 00       LDA   #00
F5F4   85 2E       STA   2E
F5F6   85 30       STA   30
F5F8   85 4F       STA   4F
F5FA   A5 31       LDA   31
F5FC   85 4E       STA   4E
F5FE   A9 01       LDA   #01
F600   85 31       STA   31
F602   85 2F       STA   2F
F604   A9 BB       LDA   #BB
F606   85 34       STA   34
F608   85 36       STA   36
F60A   20 E6 F7    JSR   F7E6
F60D   A5 52       LDA   52
F60F   85 38       STA   38
F611   A4 36       LDY   36
F613   A5 53       LDA   53
F615   91 2E       STA   (2E),Y
F617   C8          INY
F618   A5 54       LDA   54
F61A   91 2E       STA   (2E),Y
F61C   C8          INY
F61D   A5 55       LDA   55
F61F   91 2E       STA   (2E),Y
F621   C8          INY
F622   84 36       STY   36
F624   20 E6 F7    JSR   F7E6
F627   A4 36       LDY   36
F629   A5 52       LDA   52
F62B   91 2E       STA   (2E),Y
F62D   C8          INY
F62E   A5 53       LDA   53
F630   91 2E       STA   (2E),Y
F632   C8          INY
F633   F0 0E       BEQ   F643
F635   A5 54       LDA   54
F637   91 2E       STA   (2E),Y
```

```
F639  C8          INY
F63A  A5 55       LDA   55
F63C  91 2E       STA   (2E),Y
F63E  C8          INY
F63F  84 36       STY   36
F641  D0 E1       BNE   F624
F643  A5 54       LDA   54
F645  91 30       STA   (30),Y
F647  C8          INY
F648  A5 55       LDA   55
F64A  91 30       STA   (30),Y
F64C  C8          INY
F64D  84 36       STY   36
F64F  20 E6 F7    JSR   F7E6
F652  A4 36       LDY   36
F654  A5 52       LDA   52
F656  91 30       STA   (30),Y
F658  C8          INY
F659  A5 53       LDA   53
F65B  91 30       STA   (30),Y
F65D  C8          INY
F65E  A5 54       LDA   54
F660  91 30       STA   (30),Y
F662  C8          INY
F663  A5 55       LDA   55
F665  91 30       STA   (30),Y
F667  C8          INY
F668  84 36       STY   36
F66A  C0 BB       CPY   #BB
F66C  90 E1       BCC   F64F
F66E  A9 45       LDA   #45
F670  85 2E       STA   2E
F672  A5 31       LDA   31
F674  85 2F       STA   2F
F676  A0 BA       LDY   #BA
F678  B1 30       LDA   (30),Y
F67A  91 2E       STA   (2E),Y
F67C  88          DEY
F67D  D0 F9       BNE   F678
F67F  B1 30       LDA   (30),Y
F681  91 2E       STA   (2E),Y
F683  A2 BB       LDX   #BB
F685  BD 00 01    LDA   0100,X
F688  91 30       STA   (30),Y
F68A  C8          INY
F68B  E8          INX
F68C  D0 F7       BNE   F685
F68E  86 50       STX   50
F690  60          RTS

*****************************
F691  C9 20       CMP   #20      "Verify utasításkód"?
F693  F0 03       BEQ   F698     igen
F695  4C CA F6    JMP   F6CA     utasításkód ellenőrzését folytatni!

F698  20 E9 F5    JSR   F5E9     az adatpuffer paritásának kiszámítása
F69B  85 3A       STA   3A       és tárolása
F69D  20 8F F7    JSR   F78F
F6A0  20 0A F5    JSR   F50A     adatblokk kezdetének megkeresése
F6A3  A0 BB       LDY   #BB
F6A5  B9 00 01    LDA   0100,Y   adatok a pufferból
F6A8  50 FE       BVC   F6A8     Byte Ready?
F6AA  B8          CLV
F6AB  4D 01 1C    EOR   1C01     összehasonlítani e lemezről származó adatokkal!
F6AE  D0 15       BNE   F6C5     ha nem egyenő, hiba!
F6B0  C8          INY
F6B1  D0 F2       BNE   F6A5
```

```
F6B3  B1 30     LDA  (30),Y     adatok a pufferből
F6B5  50 FE     BVC   F6B5
F6B7  B8        CLV
F6B8  4D 01 1C  EOR   1C01      összehasonlitani a lemezről származó adatokkal!
F6BB  D0 08     BNE   F6C5      ha nem egyenlő, hiba!
F6BD  C8        INY
F6BE  C0 FD     CPY  #FD
F6C0  D0 F1     BNE   F6B3
F6C2  4C 18 F4  JMP   F418      hibamentes lezárás

F6C5  A9 07     LDA  #07
F6C7  4C 69 F9  JMP   F969      25, "write error"

*****************************
F6CA  20 10 F5  JSR   F510      blokk-fejléc olvasása
F6CD  4C 18 F4  JMP   F418      kész

*****************************
F6D0  A9 00     LDA  #00
F6D2  85 57     STA   57
F6D4  85 5A     STA   5A
F6D6  A4 34     LDY   34
F6D8  A5 52     LDA   52
F6DA  29 F0     AND  #F0        a felső Nibble leválasztása
F6DC  4A        LSR   A
F6DD  4A        LSR   A
F6DE  4A        LSR   A
F6DF  4A        LSR   A
F6E0  AA        TAX             mint index a táblázatban
F6E1  BD 7F F7  LDA   F77F,X
F6E4  0A        ASL   A
F6E5  0A        ASL   A         8-szor
F6E6  0A        ASL   A
F6E7  85 56     STA   56
F6E9  A5 52     LDA   52
F6EB  29 0F     AND  #0F        alsó Nibble leválasztása
F6ED  AA        TAX             mint index a táblázatban
F6EE  BD 7F F7  LDA   F77F,X
F6F1  6A        ROR   A
F6F2  66 57     ROR   57
F6F4  6A        ROR   A
F6F5  66 57     ROR   57
F6F7  29 07     AND  #07
F6F9  05 56     ORA   56
F6FB  91 30     STA  (30),Y     a pufferben
F6FD  C8        INY             a puffer mutató növelése
F6FE  A5 53     LDA   53
F700  29 F0     AND  #F0        felső Nibble leválasztása
F702  4A        LSR   A
F703  4A        LSR   A
F704  4A        LSR   A
F705  4A        LSR   A
F706  AA        TAX             indexként a táblázatban
F707  BD 7F F7  LDA   F77F,X
F70A  0A        ASL   A
F70B  05 57     ORA   57
F70D  85 57     STA   57
F70F  A5 53     LDA   53
F711  29 0F     AND  #0F        alsó Nibble
F713  AA        TAX             mint index
F714  BD 7F F7  LDA   F77F,X
F717  2A        ROL   A
F718  2A        ROL   A
F719  2A        ROL   A
F71A  2A        ROL   A
F71B  85 58     STA   58
F71D  2A        ROL   A
```

```
F71E  29 01      AND  #01
F720  05 57      ORA   57
F722  91 30      STA  (30),Y
F724  C8         INY              a puffer mutató növelése
F725  A5 54      LDA   54
F727  29 F0      AND  #F0         Hi-Nibble leválasztása
F729  4A         LSR   A
F72A  4A         LSR   A
F72B  4A         LSR   A
F72C  4A         LSR   A
F72D  AA         TAX
F72E  BD 7F F7   LDA   F77F,X
F731  18         CLC
F732  6A         ROR   A
F733  05 58      ORA   58
F735  91 30      STA  (30),Y
F737  C8         INY              a puffer mutató növelése
F738  6A         ROR   A
F739  29 80      AND  #80
F73B  85 59      STA   59
F73D  A5 54      LDA   54
F73F  29 0F      AND  #0F         alsó Nibble
F741  AA         TAX              indexként
F742  BD 7F F7   LDA   F77F,X
F745  0A         ASL   A
F746  0A         ASL   A
F747  29 7C      AND  #7C
F749  05 59      ORA   59
F74B  85 59      STA   59
F74D  A5 55      LDA   55
F74F  29 F0      AND  #F0         Hi-Nibble leválasztása
F751  4A         LSR   A
F752  4A         LSR   A
F753  4A         LSR   A
F754  4A         LSR   A
F755  AA         TAX              indexként a táblázatban
F756  BD 7F F7   LDA   F77F,X
F759  6A         ROR   A
F75A  66 5A      ROR   5A
F75C  6A         ROR   A
F75D  66 5A      ROR   5A
F75F  6A         ROR   A
F760  66 5A      ROR   5A
F762  29 03      AND  #03
F764  05 59      ORA   59
F766  91 30      STA  (30),Y
F768  C8         INY              a puffer mutató növelése
F769  D0 04      BNE   F76F
F76B  A5 2F      LDA   2F
F76D  85 31      STA   31
F76F  A5 55      LDA   55
F771  29 0F      AND  #0F         alsó Nibble
F773  AA         TAX              indexként
F774  BD 7F F7   LDA   F77F,X
F777  05 5A      ORA   5A
F779  91 30      STA  (30),Y
F77B  C8         INY              a puffer mutató növelése
F77C  84 34      STY   34         és tárolása
F77E  60         RTS

******************************
F77F  0A 0B 12 13 0E 0F 16 17
F787  09 19 1A 1B 0D 1D 1E 15

******************************
F78F  A9 00      LDA  #00
F791  85 30      STA   30
```

```
F793   85 2E      STA   2E
F795   85 36      STA   36
F797   A9 BB      LDA   #BB
F799   85 34      STA   34
F79B   85 50      STA   50
F79D   A5 31      LDA   31
F79F   85 2F      STA   2F
F7A1   A9 01      LDA   #01
F7A3   85 31      STA   31
F7A5   A5 47      LDA   47
F7A7   85 52      STA   52
F7A9   A4 36      LDY   36
F7AB   B1 2E      LDA   (2E),Y
F7AD   85 53      STA   53
F7AF   C8         INY
F7B0   B1 2E      LDA   (2E),Y
F7B2   85 54      STA   54
F7B4   C8         INY
F7B5   B1 2E      LDA   (2E),Y
F7B7   85 55      STA   55
F7B9   C8         INY
F7BA   84 36      STY   36
F7BC   20 D0 F6   JSR   F6D0
F7BF   A4 36      LDY   36
F7C1   B1 2E      LDA   (2E),Y
F7C3   85 52      STA   52
F7C5   C8         INY
F7C6   F0 11      BEQ   F7D9
F7C8   B1 2E      LDA   (2E),Y
F7CA   85 53      STA   53
F7CC   C8         INY
F7CD   B1 2E      LDA   (2E),Y
F7CF   85 54      STA   54
F7D1   C8         INY
F7D2   B1 2E      LDA   (2E),Y
F7D4   85 55      STA   55
F7D6   C8         INY
F7D7   D0 E1      BNE   F7BA
F7D9   A5 3A      LDA   3A
F7DB   85 53      STA   53
F7DD   A9 00      LDA   #00
F7DF   85 54      STA   54
F7E1   85 55      STA   55
F7E3   4C D0 F6   JMP   F6D0

F7E6   A4 34      LDY   34
F7E8   B1 30      LDA   (30),Y
F7EA   29 F8      AND   #F8
F7EC   4A         LSR   A
F7ED   4A         LSR   A
F7EE   4A         LSR   A
F7EF   85 56      STA   56
F7F1   B1 30      LDA   (30),Y
F7F3   29 07      AND   #07
F7F5   0A         ASL   A
F7F6   0A         ASL   A
F7F7   85 57      STA   57
F7F9   C8         INY
F7FA   D0 06      BNE   F802
F7FC   A5 4E      LDA   4E
F7FE   85 31      STA   31
F800   A4 4F      LDY   4F
F802   B1 30      LDA   (30),Y
F804   29 C0      AND   #C0
F806   2A         ROL   A
F807   2A         ROL   A
F808   2A         ROL   A
```

```
F809   05 57        ORA   57
F80B   85 57        STA   57
F80D   B1 30        LDA   (30),Y
F80F   29 3E        AND   #3E
F811   4A           LSR   A
F812   85 58        STA   58
F814   B1 30        LDA   (30),Y
F816   29 01        AND   #01
F818   0A           ASL   A
F819   0A           ASL   A
F81A   0A           ASL   A
F81B   0A           ASL   A
F81C   85 59        STA   59
F81E   C8           INY
F81F   B1 30        LDA   (30),Y
F821   29 F0        AND   #F0
F823   4A           LSR   A
F824   4A           LSR   A
F825   4A           LSR   A
F826   4A           LSR   A
F827   05 59        ORA   59
F829   85 59        STA   59
F82B   B1 30        LDA   (30),Y
F82D   29 0F        AND   #0F
F82F   0A           ASL   A
F830   85 5A        STA   5A
F832   C8           INY
F833   B1 30        LDA   (30),Y
F835   29 80        AND   #80
F837   18           CLC
F838   2A           ROL   A
F839   2A           ROL   A
F83A   29 01        AND   #01
F83C   05 5A        ORA   5A
F83E   85 5A        STA   5A
F840   B1 30        LDA   (30),Y
F842   29 7C        AND   #7C
F844   4A           LSR   A
F845   4A           LSR   A
F846   85 5B        STA   5B
F848   B1 30        LDA   (30),Y
F84A   29 03        AND   #03
F84C   0A           ASL   A
F84D   0A           ASL   A
F84E   0A           ASL   A
F84F   85 5C        STA   5C
F851   C8           INY
F852   D0 06        BNE   F85A
F854   A5 4E        LDA   4E
F856   85 31        STA   31
F858   A4 4F        LDY   4F
F85A   B1 30        LDA   (30),Y
F85C   29 E0        AND   #E0
F85E   2A           ROL   A
F85F   2A           ROL   A
F860   2A           ROL   A
F861   2A           ROL   A
F862   05 5C        ORA   5C
F864   85 5C        STA   5C
F866   B1 30        LDA   (30),Y
F868   29 1F        AND   #1F
F86A   85 5D        STA   5D
F86C   C8           INY
F86D   84 34        STY   34
F86F   A6 56        LDX   56
F871   BD A0 F8     LDA   F8A0,X
F874   A6 57        LDX   57
```

```
F876   1D C0 F8     ORA   F8C0,X
F879   85 52        STA   52
F87B   A6 58        LDX   58
F87D   BD A0 F8     LDA   F8A0,X
F880   A6 59        LDX   59
F882   1D C0 F8     ORA   F8C0,X
F885   85 53        STA   53
F887   A6 5A        LDX   5A
F889   BD A0 F8     LDA   F8A0,X
F88C   A6 5B        LDX   5B
F88E   1D C0 F8     ORA   F8C0,X
F891   85 54        STA   54
F893   A6 5C        LDX   5C
F895   BD A0 F8     LDA   F8A0,X
F898   A6 5D        LDX   5D
F89A   1D C0 F8     ORA   F8C0,X
F89D   85 55        STA   55
F89F   60           RTS

****************************
F8A0   FF FF FF FF FF FF FF FF
F8A8   FF 80 00 10 FF C0 40 50
F8B0   FF FF 20 30 FF F0 60 70
F8B8   FF 90 A0 B0 FF D0 E0 FF
F8C0   FF FF FF FF FF FF FF FF
F8C8   FF 08 00 01 FF 0C 04 05
F8D0   FF FF 02 03 FF 0F 06 07
F8D8   FF 09 0A 0B FF 0D 0E FF

****************************
F8E0   A9 00        LDA   #00
F8E2   85 34        STA    34
F8E4   85 2E        STA    2E
F8E6   85 36        STA    36
F8E8   A9 01        LDA   #01
F8EA   85 4E        STA    4E
F8EC   A9 BA        LDA   #BA
F8EE   85 4F        STA    4F
F8F0   A5 31        LDA    31
F8F2   85 2F        STA    2F
F8F4   20 E6 F7     JSR    F7E6
F8F7   A5 52        LDA    52
F8F9   85 38        STA    38
F8FB   A4 36        LDY    36
F8FD   A5 53        LDA    53
F8FF   91 2E        STA   (2E),Y
F901   C8           INY
F902   A5 54        LDA    54
F904   91 2E        STA   (2E),Y
F906   C8           INY
F907   A5 55        LDA    55
F909   91 2E        STA   (2E),Y
F90B   C8           INY
F90C   84 36        STY    36
F90E   20 E6 F7     JSR    F7E6
F911   A4 36        LDY    36
F913   A5 52        LDA    52
F915   91 2E        STA   (2E),Y
F917   C8           INY
F918   F0 11        BEQ    F92B
F91A   A5 53        LDA    53
F91C   91 2E        STA   (2E),Y
F91E   C8           INY
F91F   A5 54        LDA    54
F921   91 2E        STA   (2E),Y
F923   C8           INY
F924   A5 55        LDA    55
```

```
F926   91 2E      STA   (2E),Y
F928   C8         INY
F929   D0 E1      BNE   F90C
F92B   A5 53      LDA   53
F92D   85 3A      STA   3A
F92F   A5 2F      LDA   2F
F931   85 31      STA   31
F933   60         RTS

F934   A5 31      LDA   31
F936   85 2F      STA   2F
F938   A9 00      LDA   #00
F93A   85 31      STA   31
F93C   A9 24      LDA   #24
F93E   85 34      STA   34
F940   A5 39      LDA   39
F942   85 52      STA   52
F944   A5 1A      LDA   1A
F946   85 53      STA   53
F948   A5 19      LDA   19
F94A   85 54      STA   54
F94C   A5 18      LDA   18
F94E   85 55      STA   55
F950   20 D0 F6   JSR   F6D0
F953   A5 17      LDA   17
F955   85 52      STA   52
F957   A5 16      LDA   16
F959   85 53      STA   53
F95B   A9 00      LDA   #00
F95D   85 54      STA   54
F95F   85 55      STA   55
F961   20 D0 F6   JSR   F6D0
F964   A5 2F      LDA   2F
F966   85 31      STA   31
F968   60         RTS

F969   A4 3F      LDY   3F
F96B   99 00 00   STA   0000,Y
F96E   A5 50      LDA   50
F970   F0 03      BEQ   F975
F972   20 F2 F5   JSR   F5F2
F975   20 8F F9   JSR   F98F
F978   A6 49      LDX   49        a verem-mutató visszatöltése
F97A   9A         TXS
F97B   4C BE F2   JMP   F2BE

F97E   A9 A0      LDA   #A0
F980   85 20      STA   20
F982   AD 00 1C   LDA   1C00
F985   09 04      ORA   #04       a meghajtó-motor kikapcsolása
F987   8D 00 1C   STA   1C00
F98A   A9 3C      LDA   #3C
F98C   85 48      STA   48
F98E   60         RTS

F98F   A6 3E      LDX   3E
F991   A5 20      LDA   20
F993   09 10      ORA   #10
F995   85 20      STA   20
F997   A9 FF      LDA   #FF
F999   85 48      STA   48
F99B   60         RTS

F99C   AD 07 1C   LDA   1C07
F99F   8D 05 1C   STA   1C05
F9A2   AD 00 1C   LDA   1C00
F9A5   29 10      AND   #10       "Write Protect"?
```

```
F9A7  C5 1E     CMP  1E
F9A9  85 1E     STA  1E
F9AB  F0 04     BEQ  F9B1
F9AD  A9 01     LDA  #01
F9AF  85 1C     STA  1C
F9B1  AD FE 02  LDA  02FE
F9B4  F0 15     BEQ  F9CB
F9B6  C9 02     CMP  #02
F9B8  D0 07     BNE  F9C1
F9BA  A9 00     LDA  #00
F9BC  8D FE 02  STA  02FE
F9BF  F0 0A     BEQ  F9CB
F9C1  85 4A     STA  4A
F9C3  A9 02     LDA  #02
F9C5  8D FE 02  STA  02FE
F9C8  4C 2E FA  JMP  FA2E

F9CB  A6 3E     LDX  3E
F9CD  30 07     BMI  F9D6
F9CF  A5 20     LDA  20
F9D1  A8        TAY
F9D2  C9 20     CMP  #20
F9D4  D0 03     BNE  F9D9
F9D6  4C BE FA  JMP  FABE

F9D9  C6 48     DEC  48
F9DB  D0 1D     BNE  F9FA
F9DD  98        TYA
F9DE  10 04     BPL  F9E4
F9E0  29 7F     AND  #7F
F9E2  85 20     STA  20
F9E4  29 10     AND  #10
F9E6  F0 12     BEQ  F9FA
F9E8  AD 00 1C  LDA  1C00
F9EB  29 FB     AND  #FB          a meghajtóegység motorja BE
F9ED  8D 00 1C  STA  1C00
F9F0  A9 FF     LDA  #FF
F9F2  85 3E     STA  3E
F9F4  A9 00     LDA  #00
F9F6  85 20     STA  20
F9F8  F0 DC     BEQ  F9D6
F9FA  98        TYA
F9FB  29 40     AND  #40
F9FD  D0 03     BNE  FA02
F9FF  4C BE FA  JMP  FABE

FA02  6C 62 00  JMP  (0062)
FA05  A5 4A     LDA  4A
FA07  10 05     BPL  FA0E
FA09  49 FF     EOR  #FF
FA0B  18        CLC
FA0C  69 01     ADC  #01
FA0E  C5 64     CMP  64
FA10  B0 0A     BCS  FA1C
FA12  A9 3B     LDA  #3B
FA14  85 62     STA  62
FA16  A9 FA     LDA  #FA          $62/$63 mutató $FA3B-n
FA18  85 63     STA  63
FA1A  D0 12     BNE  FA2E
FA1C  E5 5E     SBC  5E
FA1E  E5 5E     SBC  5E
FA20  85 61     STA  61
FA22  A5 5E     LDA  5E
FA24  85 60     STA  60
FA26  A9 7B     LDA  #7B
FA28  85 62     STA  62
FA2A  A9 FA     LDA  #FA          $62/$63 mutató $FA7B-n
```

```
FA2C  85 63      STA  63
FA2E  A5 4A      LDA  4A     mágnesfej-pozicionálás léptetés-számlálójának
FA30  10 31      BPL  FA63
FA32  E6 4A      INC  4A     növelése
FA34  AE 00 1C   LDX  1C00
FA37  CA         DEX
FA38  4C 69 FA   JMP  FA69

*****************************
FA3B  A5 4A      LDA  4A     fejpozicionálás léptetés-számlálója
FA3D  D0 EF      BNE  FA2E   még nem nulla?
FA3F  A9 4E      LDA  #4E
FA41  85 62      STA  62
FA43  A9 FA      LDA  #FA    $62/$63 mutató $FA4E-n
FA45  85 63      STA  63
FA47  A9 05      LDA  #05
FA49  85 60      STA  60     mutató 5-ön
FA4B  4C BE FA   JMP  FABE

*****************************
FA4E  C6 60      DEC  60     számlálót csökkenteni!
FA50  D0 6C      BNE  FABE   még nem nulla?
FA52  A5 20      LDA  20
FA54  29 BF      AND  #BF    6. bit törlése
FA56  85 20      STA  20
FA58  A9 05      LDA  #05
FA5A  85 62      STA  62
FA5C  A9 FA      LDA  #FA    $62/$63 mutató $FA05-ön
FA5E  85 63      STA  63
FA60  4C BE FA   JMP  FABE

*****************************  a fejpozicionálás léptetés-számláló-
FA63  C6 4A      DEC  4A     jának csökkentése
FA65  AE 00 1C   LDX  1C00
FA68  E8         INX
FA69  8A         TXA
FA6A  29 03      AND  #03
FA6C  85 4B      STA  4B
FA6E  AD 00 1C   LDA  1C00
FA71  29 FC      AND  #FC
FA73  05 4B      ORA  4B     léptetőmotor KI
FA75  8D 00 1C   STA  1C00
FA78  4C BE FA   JMP  FABE

*****************************
FA7B  38         SEC
FA7C  AD 07 1C   LDA  1C07
FA7F  E5 5F      SBC  5F
FA81  8D 05 1C   STA  1C05
FA84  C6 60      DEC  60     számlálót csökkenteni!
FA86  D0 0C      BNE  FA94   még nem nulla?
FA88  A5 5E      LDA  5E
FA8A  85 60      STA  60     számlálót újra beállítani!
FA8C  A9 97      LDA  #97
FA8E  85 62      STA  62
FA90  A9 FA      LDA  #FA    $62/$63 mutató $FA97-en
FA92  85 63      STA  63
FA94  4C 2E FA   JMP  FA2E

*****************************
FA97  C6 61      DEC  61
FA99  D0 F9      BNE  FA94
FA9B  A9 A5      LDA  #A5
FA9D  85 62      STA  62
FA9F  A9 FA      LDA  #FA    $62/$63 mutató $FAA5-ön
FAA1  85 63      STA  63
FAA3  D0 EF      BNE  FA94
```

```
******************************
FAA5  AD 07 1C    LDA   1C07
FAA8  18          CLC
FAA9  65 5F       ADC   5F
FAAB  8D 05 1C    STA   1C05
FAAE  C6 60       DEC   60        a számláló csökkentése
FAB0  D0 E2       BNE   FA94      még nem nulla?
FAB2  A9 4E       LDA   #4E
FAB4  85 62       STA   62
FAB6  A9 FA       LDA   #FA       $62/$63 mutató $FA4E-n
FAB8  85 63       STA   63
FABA  A9 05       LDA   #05
FABC  85 60       STA   60        mutató 5-ön
FABE  AD 0C 1C    LDA   1C0C
FAC1  29 FD       AND   #FD       az 1. bit törlése
FAC3  8D 0C 1C    STA   1C0C
FAC6  60          RTS

******************************    formálás
FAC7  A5 51       LDA   51        sáv-szám
FAC9  10 2A       BPL   FAF5      folyamatban van már a formálás?
FACB  A6 3D       LDX   3D        egységszám
FACD  A9 60       LDA   #60       fejpozicionálási kapcsoló
FACF  95 20       STA   20,X      beállítása
FAD1  A9 01       LDA   #01
FAD3  95 22       STA   22,X      célsáv beállítása
FAD5  85 51       STA   51        sáv-sorszám a formáláskor
FAD7  A9 A4       LDA   #A4       164
FAD9  85 4A       STA   4A        fejpozicionálás léptetés-számlálója
FADB  AD 00 1C    LDA   1C00
FADE  29 FC       AND   #FC       léptetőmotor BE
FAE0  8D 00 1C    STA   1C00
FAE3  A9 0A       LDA   #0A       10
FAE5  8D 20 06    STA   0620      hiba-számláló
FAE8  A9 A0       LDA   #A0       $621/$622 = 4000
FAEA  8D 21 06    STA   0621      sávkapacitás meghat. miatt inicializálás!
FAED  A9 0F       LDA   #0F       4000   kapacitás    2*4000 byte
FAEF  8D 22 06    STA   0622
FAF2  4C 9C F9    JMP   F99C      vissza job-ciklusba

FAF5  A0 00       LDY   #00
FAF7  D1 32       CMP   (32),Y
FAF9  F0 05       BEQ   FB00
FAFB  91 32       STA   (32),Y
FAFD  4C 9C F9    JMP   F99C      job-ciklushoz

FB00  AD 00 1C    LDA   1C00
FB03  29 10       AND   #10       "Write Protect"?
FB05  D0 05       BNE   FB0C      nem
FB07  A9 08       LDA   #08
FB09  4C D3 FD    JMP   FDD3      26, "write protect on"

FB0C  20 A3 FD    JSR   FDA3      a $FF kód felírása lemezre 10240-szer
FB0F  20 C3 FD    JSR   FDC3      /$621/$622/-szer a $FF kód felírása a lemezre
FB12  A9 55       LDA   #55       $55
FB14  8D 01 1C    STA   1C01
FB17  20 C3 FD    JSR   FDC3      és /$621/$622/-szer a lemezre
FB1A  20 00 FE    JSR   FE00      olvasásra átkapcsolni
FB1D  20 56 F5    JSR   F556      óra bekapcsolása, $FF /SYNC/ megkeresése
FB20  A9 40       LDA   #40
FB22  0D 0B 18    ORA   180B      Timer 1
FB25  8D 0B 18    STA   180B
FB28  A9 62       LDA   #62       98 ütemciklus, kb. 0,1 ms
FB2A  8D 06 18    STA   1806
FB2D  A9 00       LDA   #00
FB2F  8D 07 18    STA   1807
FB32  8D 05 18    STA   1805      az Óra indítása
```

```
FB35  A0 00     LDY  #00      számláló nullán
FB37  A2 00     LDX  #00
FB39  2C 00 1C  BIT  1C00     megvan a SYNC?
FB3C  30 FB     BMI  FB39     nincs várni!
FB3E  2C 00 1C  BIT  1C00     megvan a SYNC?
FB41  10 FB     BPL  FB3E     megvárni a SYNC-tartomány végét!
FB43  AD 04 18  LDA  1804     az óra megszakítás-kapcsoló visszaállítása
FB46  2C 00 1C  BIT  1C00     megvan a SYNC?
FB49  10 11     BPL  FB5C     nincs vége a SYNC tartománynak /$55/?
FB4B  AD 0D 18  LDA  180D     megszakítás-kapcsoló regiszter
FB4E  0A        ASL  A
FB4F  10 F5     BPL  FB46     az óra még nem járt le
FB51  E8        INX           számláló növelése
FB52  D0 EF     BNE  FB43
FB54  C8        INY           számláló Hi-byte-jának növelése
FB55  D0 EC     BNE  FB43
FB57  A9 02     LDA  #02      ha tulcsordulás, hiba!
FB59  4C D3 FD  JMP  FDD3     20, "read error"

FB5C  86 71     STX  71       a számláló állása = a $55 tartomány időtartama
FB5E  84 72     STY  72       tárolás
FB60  A2 00     LDX  #00
FB62  A0 00     LDY  #00      a számláló ismét nullán
FB64  AD 04 18  LDA  1804     a Timer 1 megszakítás-kapcsoló visszaállítása
FB67  2C 00 1C  BIT  1C00     megvan a SYNC?
FB6A  30 11     BMI  FB7D     igen
FB6C  AD 0D 18  LDA  180D     megszakítás-kapcsoló regiszter
FB6F  0A        ASL  A        óra-kapcsoló a 7. bit után!
FB70  10 F5     BPL  FB67     nem, megvárni az óra lefutását
FB72  E8        INX
FB73  D0 EF     BNE  FB64     a számláló növelése
FB75  C8        INY
FB76  D0 EC     BNE  FB64
FB78  A9 02     LDA  #02      ha tulcsordulás, hiba
FB7A  4C D3 FD  JMP  FDD3     20, "read error"

FB7D  38        SEC
FB7E  8A        TXA
FB7F  E5 71     SBC  71       a számláló állás /$55/
FB81  AA        TAX
FB82  85 70     STA  70
FB84  98        TYA           és a $FF-tartomány értéke közötti
                              különbséget
FB85  E5 72     SBC  72
FB87  A8        TAY           $70/$71 mögé vinni!
FB88  85 71     STA  71
FB8A  10 0B     BPL  FB97     pozitiv különbség?
FB8C  49 FF     EOR  #FF
FB8E  A8        TAY
FB8F  8A        TXA
FB90  49 FF     EOR  #FF      a különbség abszolút értékének kiszámítása
FB92  AA        TAX
FB93  E8        INX
FB94  D0 01     BNE  FB97
FB96  C8        INY
FB97  98        TYA
FB98  D0 04     BNE  FB9E
FB9A  E0 04     CPX  #04      4 * 0,1 ms-nál   kisebb a különbség?
FB9C  90 18     BCC  FBB6     igen
FB9E  06 70     ASL  70
FBA0  26 71     ROL  71       megkétszerezni a különbséget!
FBA2  18        CLC
FBA3  A5 70     LDA  70
FBA5  6D 21 06  ADC  0621
FBA8  8D 21 06  STA  0621     hozzáadni a 4000 kiindulási értékhez!
FBAB  A5 71     LDA  71
FBAD  6D 22 06  ADC  0622
```

```
FBB0  8D 22 06   STA   0622
FBB3  4C 0C FB   JMP   FB0C      ismételni, amig a különbség Ø,4 ms-nál
                                 kisebb lesz!
FBB6  A2 00      LDX   #00
FBB8  A0 00      LDY   #00       számláló ismét nullán
FBBA  B8         CLV
FBBB  AD 00 1C   LDA   1C00      SYNC?
FBBE  10 0E      BPL   FBCE      nem
FBC0  50 F9      BVC   FBBB      Byte Ready?
FBC2  B8         CLV
FBC3  E8         INX
FBC4  D0 F5      BNE   FBBB      a számláló növelése
FBC6  C8         INY
FBC7  D0 F2      BNE   FBBB
FBC9  A9 03      LDA   #03       ha túlcsordulás, hiba
FBCB  4C D3 FD   JMP   FDD3      21, "read error"

FBCE  8A         TXA
FBCF  0A         ASL   A         a számláló megkétszerezése
FBD0  8D 25 06   STA   0625
FBD3  98         TYA
FBD4  2A         ROL   A         és a $624/$625 után, mint sáv-kapacitás
FBD5  8D 24 06   STA   0624
FBD8  A9 BF      LDA   #BF
FBDA  2D 0B 18   AND   180B
FBDD  8D 0B 18   STA   180B
FBE0  A9 66      LDA   #66       1Ø2
FBE2  8D 26 06   STA   0626
FBE5  A6 43      LDX   43        szektorok száma ezen a sávon
FBE7  A0 00      LDY   #00
FBE9  98         TYA
FBEA  18         CLC
FBEB  6D 26 06   ADC   0626
FBEE  90 01      BCC   FBF1
FBF0  C8         INY
FBF1  C8         INY
FBF2  CA         DEX
FBF3  D0 F5      BNE   FBEA      blokk-közökben lévő összes byte
FBF5  49 FF      EOR   #FF       számának kiszámítása
FBF7  38         SEC
FBF8  69 00      ADC   #00
FBFA  18         CLC
FBFB  6D 25 06   ADC   0625
FBFE  B0 03      BCS   FC03
FC00  CE 24 06   DEC   0624
FC03  AA         TAX
FC04  98         TYA
FC05  49 FF      EOR   #FF
FC07  38         SEC
FC08  69 00      ADC   #00
FC0A  18         CLC
FC0B  6D 24 06   ADC   0624      eredmény az A/X-ben
FC0E  10 05      BPL   FC15
FC10  A9 04      LDA   #04
FC12  4C D3 FD   JMP   FDD3      22, "read error"

FC15  A8         TAY
FC16  8A         TXA
FC17  A2 00      LDX   #00
FC19  38         SEC             az összeg osztva
FC1A  E5 43      SBC   43        a szektorok számával /$43/
FC1C  B0 03      BCS   FC21
FC1E  88         DEY
FC1F  30 03      BMI   FC24
FC21  E8         INX
FC22  D0 F5      BNE   FC19
FC24  8E 26 06   STX   0626      blokk-közönkénti byte-ok számának
```

```
FC27  E0 04     CPX  #04      összehasonlítása a minimális értékkel
FC29  B0 05     BCS  FC30     ok
FC2B  A9 05     LDA  #05
FC2D  4C D3 FD  JMP  FDD3     23, "read error"

FC30  18        CLC           osztás maradéka
FC31  65 43     ADC  43       plusz a szektorok számának
FC33  8D 27 06  STA  0627     tárolása
FC36  A9 00     LDA  #00
FC38  8D 28 06  STA  0628     szektor-számláló
FC3B  A0 00     LDY  #00      számláló /Lo/
FC3D  A6 3D     LDX  3D       egységszám
FC3F  A5 39     LDA  39       8 állandó, fejléc kezdetének jele
FC41  99 00 03  STA  0300,Y   a pufferben
FC44  C8        INY
FC45  C8        INY
FC46  AD 28 06  LDA  0628     szektor-szám
FC49  99 00 03  STA  0300,Y   a pufferben
FC4C  C8        INY
FC4D  A5 51     LDA  51       sáv-szám
FC4F  99 00 03  STA  0300,Y   a pufferben
FC52  C8        INY
FC53  B5 13     LDA  13,X     ID 2
FC55  99 00 03  STA  0300,Y   a pufferben
FC58  C8        INY
FC59  B5 12     LDA  12,X     ID 1
FC5B  99 00 03  STA  0300,Y   a pufferben
FC5E  C8        INY
FC5F  A9 0F     LDA  #0F      15
FC61  99 00 03  STA  0300,Y   a pufferben
FC64  C8        INY
FC65  99 00 03  STA  0300,Y   15 a pufferben
FC68  C8        INY
FC69  A9 00     LDA  #00
FC6B  59 FA 02  EOR  02FA,Y
FC6E  59 FB 02  EOR  02FB,Y
FC71  59 FC 02  EOR  02FC,Y   ellenőrzőösszeg képzése
FC74  59 FD 02  EOR  02FD,Y
FC77  99 F9 02  STA  02F9,Y
FC7A  EE 28 06  INC  0628     a számláló növelése
FC7D  AD 28 06  LDA  0628     számláló
FC80  C5 43     CMP  43       összehasonlítása a szektorok számával
FC82  90 BB     BCC  FC3F     ha kisebb, folytatni
FC84  98        TYA
FC85  48        PHA
FC86  E8        INX
FC87  8A        TXA
FC88  9D 00 05  STA  0500,X
FC8B  E8        INX
FC8C  D0 FA     BNE  FC88
FC8E  A9 03     LDA  #03      puffer-mutató $300-on
FC90  85 31     STA  31
FC92  20 30 FE  JSR  FE30
FC95  68        PLA
FC96  A8        TAY
FC97  88        DEY
FC98  20 E5 FD  JSR  FDE5     a puffer-adatok másolása
FC9B  20 F5 FD  JSR  FDF5
FC9E  A9 05     LDA  #05
FCA0  85 31     STA  31       puffer-mutató $500
FCA2  20 E9 F5  JSR  F5E9     az adat-puffer paritásának kiszámítása
FCA5  85 3A     STA  3A       és tárolása
FCA7  20 8F F7  JSR  F78F
FCAA  A9 00     LDA  #00
FCAC  85 32     STA  32
FCAE  20 0E FE  JSR  FE0E     átkapcsolás írásra, $55 felírása 10240-szer!
FCB1  A9 FF     LDA  #FF
```

```
FCB3  8D 01 1C   STA  1C01
FCB6  A2 05      LDX  #05      ØFF felírása 5-ször
FCB8  50 FE      BVC  FCB8     Byte Ready?
FCBA  B8         CLV
FCBB  CA         DEX
FCBC  D0 FA      BNE  FCB8
FCBE  A2 0A      LDX  #0A      1Ø-szer
FCC0  A4 32      LDY  32       puffer-mutató
FCC2  50 FE      BVC  FCC2     Byte Ready?
FCC4  B8         CLV
FCC5  B9 00 03   LDA  0300,Y   adatok felírása
FCC8  8D 01 1C   STA  1C01     a pufferból
FCCB  C8         INY
FCCC  CA         DEX           már megtörtént 1Ø adat írása?
FCCD  D0 F3      BNE  FCC2
FCCF  A2 09      LDX  #09      9-szer
FCD1  50 FE      BVC  FCD1     Byte Ready?
FCD3  B8         CLV
FCD4  A9 55      LDA  #55      Ø55
FCD6  8D 01 1C   STA  1C01     felírása
FCD9  CA         DEX
FCDA  D0 F5      BNE  FCD1     9-szer
FCDC  A9 FF      LDA  #FF      ØFF
FCDE  A2 05      LDX  #05      5-ször
FCE0  50 FE      BVC  FCE0     Byte Ready?
FCE2  B8         CLV
FCE3  8D 01 1C   STA  1C01     írófejhez
FCE6  CA         DEX
FCE7  D0 F7      BNE  FCE0
FCE9  A2 BB      LDX  #BB
FCEB  50 FE      BVC  FCEB
FCED  B8         CLV
FCEE  BD 00 01   LDA  0100,X   Ø1BB-tól Ø1FF-ig terjedő tartomány
FCF1  8D 01 1C   STA  1C01     felírása
FCF4  E8         INX
FCF5  D0 F4      BNE  FCEB
FCF7  A0 00      LDY  #00
FCF9  50 FE      BVC  FCF9     Byte Ready?
FCFB  B8         CLV
FCFC  B1 30      LDA  (30),Y   256 adatbyte
FCFE  8D 01 1C   STA  1C01     felírása a lemezre
FD01  C8         INY
FD02  D0 F5      BNE  FCF9
FD04  A9 55      LDA  #55      Ø55
FD06  AE 26 06   LDX  0626     /Ø626/-szor
FD09  50 FE      BVC  FD09
FD0B  B8         CLV
FD0C  8D 01 1C   STA  1C01     felírása
FD0F  CA         DEX
FD10  D0 F7      BNE  FD09
FD12  A5 32      LDA  32
FD14  18         CLC
FD15  69 0A      ADC  #0A      plusz 1Ø
FD17  85 32      STA  32
FD19  CE 28 06   DEC  0628     a szektor-szám csökkentése
FD1C  D0 93      BNE  FCB1
FD1E  50 FE      BVC  FD1E     Byte Ready?
FD20  B8         CLV
FD21  50 FE      BVC  FD21     Byte Ready?
FD23  B8         CLV
FD24  20 00 FE   JSR  FE00     átkapcsolás olvasásra
FD27  A9 C8      LDA  #C8      2ØØ
FD29  8D 23 06   STA  0623
FD2C  A9 00      LDA  #00
FD2E  85 30      STA  30
FD30  A9 03      LDA  #03          puffer-mutató Ø3ØØ-on
FD32  85 31      STA  31
```

```
FD34   A5 43       LDA  43        sávonkénti szektorok száma
FD36   8D 28 06    STA  0628
FD39   20 56 F5    JSR  F556      megvárni a SYNC-et!
FD3C   A2 0A       LDX  #0A       10 adat
FD3E   A0 00       LDY  #00
FD40   50 FE       BVC  FD40      Byte Ready?
FD42   B8          CLV
FD43   AD 01 1C    LDA  1C01      byte olvasása
FD46   D1 30       CMP  (30),Y    összehasonlítás a pufferben lévő adatokkal
FD48   D0 0E       BNE  FD58      ha nem egyenlő, hiba
FD4A   C8          INY
FD4B   CA          DEX
FD4C   D0 F2       BNE  FD40
FD4E   18          CLC
FD4F   A5 30       LDA  30
FD51   69 0A       ADC  #0A       mutató növelése 10-zel
FD53   85 30       STA  30
FD55   4C 62 FD    JMP  FD62

FD58   CE 23 06    DEC  0623      számláló csökkentése
FD5B   D0 CF       BNE  FD2C      még nem nulla?
FD5D   A9 06       LDA  #06       ha nem, hiba
FD5F   4C D3 FD    JMP  FDD3      24, "read error"

FD62   20 56 F5    JSR  F556      megvárni a SYNC-et!
FD65   A0 BB       LDY  #BB
FD67   50 FE       BVC  FD67      Byte Ready?
FD69   B8          CLV
FD6A   AD 01 1C    LDA  1C01      egy byte olvasása
FD6D   D9 00 01    CMP  0100,Y    és összehasonlítása a puffer tartalmával
FD70   D0 E6       BNE  FD58      ha nem egyenlő, hiba
FD72   C8          INY
FD73   D0 F2       BNE  FD67      következő byte
FD75   A2 FC       LDX  #FC
FD77   50 FE       BVC  FD77      Byte Ready?
FD79   B8          CLV
FD7A   AD 01 1C    LDA  1C01      byte olvasása
FD7D   D9 00 05    CMP  0500,Y    és összehasonlítása a puffer tartalmával
FD80   D0 D6       BNE  FD58      ha nem egyenlő, hiba
FD82   C8          INY
FD83   CA          DEX            következő byte
FD84   D0 F1       BNE  FD77
FD86   CE 28 06    DEC  0628      szektor-számláló csökkentése
FD89   D0 AE       BNE  FD39      még nem nulla?
FD8B   E6 51       INC  51        sáv-szám növelése
FD8D   A5 51       LDA  51
FD8F   C9 24       CMP  #24       összehasonlítás
FD91   B0 03       BCS  FD96      ha nagyobb, a formálás kész
FD93   4C 9C F9    JMP  F99C      folytatás

FD96   A9 FF       LDA  #FF
FD98   85 51       STA  51        sáv-szám $FF-en
FD9A   A9 00       LDA  #00
FD9C   85 50       STA  50
FD9E   A9 01       LDA  #01
FDA0   4C 69 F9    JMP  F969      ok

******************************
FDA3   AD 0C 1C    LDA  1C0C      $FF felírása 10240-szer
FDA6   29 1F       AND  #1F       PCR átkapcsolása írásra
FDA8   09 C0       ORA  #C0
FDAA   8D 0C 1C    STA  1C0C
FDAD   A9 FF       LDA  #FF
FDAF   8D 03 1C    STA  1C03      A-kapu /író/olvasófej/ outputon
FDB2   8D 01 1C    STA  1C01      $FF felírása a lemezre
FDB5   A2 28       LDX  #28       40
FDB7   A0 00       LDY  #00
```

```
FDB9   50 FE      BVC   FDB9        Byte Ready?
FDBB   B8         CLV
FDBC   88         DEY
FDBD   D0 FA      BNE   FDB9
FDBF   CA         DEX
FDC0   D0 F7      BNE   FDB9
FDC2   60         RTS

******************************      /$621/$622/-szer írni/olvasni
FDC3   AE 21 06   LDX   0621
FDC6   AC 22 06   LDY   0622
FDC9   50 FE      BVC   FDC9        Byte Ready?
FDCB   B8         CLV
FDCC   CA         DEX
FDCD   D0 FA      BNE   FDC9
FDCF   88         DEY
FDD0   10 F7      BPL   FDC9
FDD2   60         RTS

******************************      kisérlet-számláló formálásnál
FDD3   CE 20 06   DEC   0620        a kisérletek számának csökkentése
FDD6   F0 03      BEQ   FDDB        ha nulla, hibát jelezni!
FDD8   4C 9C F9   JMP   F99C        folytatás

FDDB   A0 FF      LDY   #FF
FDDD   84 51      STY   51
FDDF   C8         INY
FDE0   84 50      STY   50
FDE2   4C 69 F9   JMP   F969        lezárás

******************************
FDE5   B9 00 03   LDA   0300,Y
FDE8   99 45 03   STA   0345,Y
FDEB   88         DEY               a puffer tartalmának másolása
FDEC   D0 F7      BNE   FDE5
FDEE   AD 00 03   LDA   0300
FDF1   8D 45 03   STA   0345
FDF4   60         RTS

******************************
FDF5   A0 44      LDY   #44
FDF7   B9 BB 01   LDA   01BB,Y      $1BB - $1FF-t
FDFA   91 30      STA   (30),Y      beírni a $30/$31 pufferbe!
FDFC   88         DEY
FDFD   10 F8      BPL   FDF7
FDFF   60         RTS

******************************      átkapcsolás olvasásra
FE00   AD 0C 1C   LDA   1C0C
FE03   09 E0      ORA   #E0         PCR átkapcsolása olvasásra
FE05   8D 0C 1C   STA   1C0C
FE08   A9 00      LDA   #00
FE0A   8D 03 1C   STA   1C03        A-kapu bemeneten
FE0D   60         RTS

******************************      $55 felírása 10240-szer
FE0E   AD 0C 1C   LDA   1C0C
FE11   29 1F      AND   #1F
FE13   09 C0      ORA   #C0         PCR átkapcsolása írásra
FE15   8D 0C 1C   STA   1C0C
FE18   A9 FF      LDA   #FF
FE1A   8D 03 1C   STA   1C03        A-kapu írófejhez irányuló outputon
FE1D   A9 55      LDA   #55         %01010101
FE1F   8D 01 1C   STA   1C01        írófejhez közvetítő A-kapun
FE22   A2 28      LDX   #28
FE24   A0 00      LDY   #00
FE26   50 FE      BVC   FE26        kész az íróelektronika byte-ja?
```

```
FE28  B8          CLV
FE29  88          DEY
FE2A  D0 FA       BNE   FE26      1Ø24Ø-szer!
FE2C  CA          DEX
FE2D  D0 F7       BNE   FE26
FE2F  60          RTS

*****************************
FE30  A9 00       LDA   #00
FE32  85 30       STA   30
FE34  85 2E       STA   2E
FE36  85 36       STA   36
FE38  A9 BB       LDA   #BB
FE3A  85 34       STA   34
FE3C  A5 31       LDA   31
FE3E  85 2F       STA   2F
FE40  A9 01       LDA   #01
FE42  85 31       STA   31
FE44  A4 36       LDY   36
FE46  B1 2E       LDA   (2E),Y
FE48  85 52       STA   52
FE4A  C8          INY
FE4B  B1 2E       LDA   (2E),Y
FE4D  85 53       STA   53
FE4F  C8          INY
FE50  B1 2E       LDA   (2E),Y
FE52  85 54       STA   54
FE54  C8          INY
FE55  B1 2E       LDA   (2E),Y
FE57  85 55       STA   55
FE59  C8          INY
FE5A  F0 08       BEQ   FE64
FE5C  84 36       STY   36
FE5E  20 D0 F6    JSR   F6D0
FE61  4C 44 FE    JMP   FE44

FE64  4C D0 F6    JMP   F6D0

*****************************     megszakítás rutin
FE67  48          PHA
FE68  8A          TXA
FE69  48          PHA             a regiszter tárolása
FE6A  98          TYA
FE6B  48          PHA
FE6C  AD 0D 18    LDA   180D      a soros busz megszakítása /ATN IN/?
FE6F  29 02       AND   #02
FE71  F0 03       BEQ   FE76      nem
FE73  20 53 E8    JSR   E853      kiszolgálni a soros buszt!
FE76  AD 0D 1C    LDA   1C0D      az 1. óra megszakítása?
FE79  0A          ASL   A
FE7A  10 03       BPL   FE7F      nem
FE7C  20 B0 F2    JSR   F2B0      a lemezvezérlő IRQ-rutinja
FE7F  68          PLA
FE80  A8          TAY
FE81  68          PLA
FE82  AA          TAX             a regiszter visszatöltése
FE83  68          PLA
FE84  40          RTI

*****************************     lemezformátum állandói
FE85  12                          18, BAM és a tartalomjegyzék sávja
FE86  04                          BAM indítása a 4. poziciótól,
FE87  04 90                       minden sávra nézve 4 byte a BAM-ban
                                  $9Ø=144, BAM vége, lemeznév kezdete
*****************************     utasítás-szavak táblázata
FE89  56 49 44 4D 42 55           "V", "I", "D", "M", "B", "U"
FE8F  50 26 43 52 53 4E           "P", "&", "C", "R", "S", "N"
```

```
*****************************        utasítás címeinek alsó byte-ja /Lo/
FE95  84 05 C1 F8 1B 5C
FE9B  07 A3 F0 88 23 0D

*****************************        utasítás címeinek felső byte-ja /Hi/
FEA1  ED D0 C8 CA CC CB
FEA7  E2 E7 C8 CA C8 EE

*****************************
FEAD  51 DD 1C 9E 1C                 szintaxis-ellenőrzési byte-ok

*****************************        file-üzemmódok
FEB2  52 57 41 4D                    "R", "W", "A", "M"

*****************************        file-tipusok
FEB6  44 53 50 55 4C                 "D", "S", "P", "U", "L"

*****************************        file-tipusok nevei
FEBB  44 53 50 55 52                 file-tipus 1. betűje "D","S","P","U","R"
FEC0  45 45 52 53 45                 file-tipus 2. betűje "E","E","R","S","E"
FEC5  4C 51 47 52 4C                 file-tipus 3. betűje "L","Q","G","R","L"

*****************************
FECA  08 00 00

*****************************
FECD  3F 7F BF FF                    a bit-utasítás maszkolása

*****************************        sávonkénti szektorok száma
FED1  11 12 13 15                    17, 18, 19, 21

*****************************        a lemezformátum állandói
FED5  41                             "A" jel az 1541 formátumra vonatkozik
FED6  04 24                          4 sáv-szám, 36, legnagyobb sáv-szám + 1
FED8  1F 19 12                       szektor-szám váltás 31, 25, 18 sáv

*****************************
FEDB  01 FF FF 01 00                 mágnesfej pozicionálásának vezérlő-byte-jai

*****************************        puffer-tár címei
FEE0  03 04 05 06 07                 /Hi/

*****************************
FEE5  07 3E

*****************************             UI-utasítástól
FEE7  6C 65 00    JMP   (0065)

*****************************        diagnosztika-rutin
FEEA  8D 00 1C    STA    1C00        LED bekapcsolása
FEED  8D 02 1C    STA    1C02        kapu outputon
FEF0  4C 7D EA    JMP    EA7D        vissza a diagnosztika-rutinhoz

*****************************        soros busz késleltető ciklusa
FEF3  8A          TXA
FEF4  A2 05       LDX   #05
FEF6  CA          DEX                kb. 40 mikroszekundum
FEF7  D0 FD       BNE    FEF6
FEF9  AA          TAX
FEFA  60          RTS

*****************************        adat-output a soros buszra
FEFB  20 AE E9    JSR    E9AE        CLOCK OUT /Hi/
FEFE  4C 9C E9    JMP    E99C        DATA OUT /Lo/
```

```
*****************************          UI-vektor
FF01  AD 02 02    LDA   0202
FF04  C9 2D       CMP  #2D
FF06  F0 05       BEQ   FF0D
FF08  38          SEC
FF09  E9 2B       SBC  #2B
FF0B  D0 DA       BNE   FEE7           /$65/ közvetett kihagyása /átugrása/
FF0D  85 23       STA   23
FF0F  60          RTS

*****************************
FF10  8E 03 18    STX   1803
FF13  A9 02       LDA  #02
FF15  8D 00 18    STA   1800
FF18  A9 1A       LDA  #1A
FF1A  8D 02 18    STA   1802
FF1D  4C A7 EA    JMP   EAA7

FF20  AD 00 18    LDA   1800
FF23  29 01       AND  #01
FF25  D0 F9       BNE   FF20
FF27  A9 01       LDA  #01
FF29  8D 05 18    STA   1805
FF2C  4C DF E9    JMP   E9DF

*****************************
FF2F  AA AA AA AA AA AA AA AA AA AA
FF39  AA AA AA AA AA AA AA AA AA AA
FF43  AA AA AA AA AA AA AA AA AA AA
FF4D  AA AA AA AA AA AA AA AA AA AA
FF57  AA AA AA AA AA AA AA AA AA AA
FF61  AA AA AA AA AA AA AA AA AA AA
FF6B  AA AA AA AA AA AA AA AA AA AA
FF75  AA AA AA AA AA AA AA AA AA AA
FF7F  AA AA AA AA AA AA AA AA AA AA
FF89  AA AA AA AA AA AA AA AA AA AA
FF93  AA AA AA AA AA AA AA AA AA AA
FF9D  AA AA AA AA AA AA AA AA AA AA
FFA7  AA AA AA AA AA AA AA AA AA AA
FFB1  AA AA AA AA AA AA AA AA AA AA
FFBB  AA AA AA AA AA AA AA AA AA AA
FFC5  AA AA AA AA AA AA AA AA AA AA
FFCF  AA AA AA AA AA AA AA AA AA AA
FFD9  AA AA AA AA AA AA AA AA AA

*****************************
FFE2  AA AA AA AA
FFE6  C6 C8 8F F9

*****************************
FFEA  5F CD
FFEC  97 CD
FFEE  00 05
FFF0  03 05
FFF2  06 05
FFF4  09 05
FFF6  0C 05
FFF8  0F 05
FFFA  01 FF

*****************************
FFFC  A0 EA
FFFE  67 FE

*****************************
```

Az eredeti könyvben szereplő és az általunk közölt DOS lista között helyenként eltérés lehet. A COMMODORE cég ugyanis időnként kisebb módosításokat végez az 1541-es lemezegység operációs rendszerében. Az általunk ismertetett DOS a jelenleg Magyarországon forgalomban lévő lemezegység operációs rendszere (a ford. megj.).

4. FEJEZET
PROGRAMOK ÉS ÖTLETEK A VC-1541-HEZ

4.1. Segédprogramok

4.1.1. Az összes file-paraméter kiírása

A lemez kilistázott tartalomjegyzéke nem tartalmazza az összes információt, amit egy-egy file-ról a DOS nyilvántart. Pedig gyakran előfordul, hogy a programozási munka során szükségünk van bizonyos adatokra, pl. a file kezdőcímére, vagy egy relatív file rekordhosszúságára stb. Ezeknek a paramétereknek a meghatározása többnyire egy-egy program megírását igényli, mint ahogyan azt a korábbi fejezetekben láttuk.
Az alábbiakban bemutatunk egy olyan programot, amely a lemezen tárolt file-ok összes paraméterét kiírja. A file-paraméterek természetesen a file-típustól is függenek. Így például egy relatív file-hoz nem lehet hozzárendelni kezdőcímet. A következő táblázat az egyes file-típusok azon paramétereit foglalja össze, amelyek a közölt programmal meghatározhatók:

PARAMÉTER	FILE-TÍPUS				
	DEL	SEQ	PRG	USR	REL
Lezárt file?	X	X	X	X	X
Védett file?	X	X	X	X	X
Foglalt blokkok száma	X	X	X	X	X
Rekordhosszúság					X
Oldalszektor-blokkok					X
Adatblokkok					X
Rekordok					X
Kezdőcím			X		
A lemez szabad blokkjai	X	X	X	X	X
A lemez foglalt blokkjai	X	X	X	X	X

A program dokumentálása a legkisebb részletekre is kiterjed. A felhasznált változók jelentését előfordulásuk sorrendjében ismertetjük.

Numerikus változók

T	– A tartalomjegyzékben lévő file-bejegyzés aktuális blokkjának sávja
S	– A tartalomjegyzékben lévő file-bejegyzés aktuális blokkjának szektora
FL	– Kapcsoló, melynek értéke eldönti, hogy a lemezről olvasott file-neveket ki kell listázni vagy összehasonlítani egy keresett file-névvel
TY	– A megadott file típusa (a bejegyzés Ø. byte-ja)
FT	– A file-típus félbyte-ja (Ø–3. bitek)
LB	– A lemezről olvasott kezdőcím alsó byte-ja

HB	– A lemezről olvasott kezdőcím felső byte-ja
BL	– A file által lefoglalt blokkok száma
RL	– A relatív file rekordhosszúsága
DT	– Egy programfile első adatblokkjának sávja, amely a kezdőcímet tartalmazza
DS	– Egy programfile első adatblokkjának szektora
AA	– Egy programfile kezdőcíme
BF	– A szabad blokkok száma a lemezen
BB	– A foglalt blokkok száma a lemezen
BS	– Az oldalszektorok száma egy relatív file-ban
RC	– A rekordok száma egy relatív file-ban

Füzérek

F$	– A keresett file neve
FF$	– Az aktuális file-név
FT$	– File-típus (szöveg)
GE$	– Az az állandó, amely megadja, hogy a file lezárt állapotban van-e (IGEN-t vagy NEM-et tartalmaz)
SA$	– Az az állandó, amely meghatározza, hogy a file védett vagy sem (IGEN-t vagy NEM-et tartalmaz)
RE$	– CHR$ (18) REVERSE ON-t tartalmaz
RA$	– CHR$ (146) REVERSE OFF-ot tartalmaz

A program dokumentációja

110	Beállítja a képernyő színét
120–200	Programfejléc
210–230	Megkérdezi a felhasználótól, hogy kéri-e a file-ok nevét. Az FL kapcsolót 1-re állítja és a 280–490-es rutinokat futtatja
250–270	A file-név bevitele. Új input szükséges, ha a file-név 16 karakternél hosszabb
280–490	A file-nevet olvassa a tartalomjegyzékből és vagy kiírja (FL = 1), vagy összehasonlítja a keresett file nevével
500–530	Beolvassa a keresett file-bejegyzés 0. byte-ját (file-típus) és a TY-ban tárolja. A jobboldali félbyte-ot tárolja az FT-ben.
540–590	A file-típust ellenőrzi, szövegesen tárolja az FT$-ban, az érvénytelen file-típust kiszűri.
600–610	A file-típus-byte 7. bitjét ellenőrzi (a file lezárt állapotban?) és az eredményt a GE$-ban tárolja.
620–630	A file-típus-byte 6. bitjét ellenőrzi (védett a file?) és az eredményt az SA$-ban tárolja.
640–690	A file által lefoglalt blokkok számát olvassa a 28. és a 29. byte-ból, majd a BL-ben tárolja.
700–730	Relatív file esetén a bejegyzés 21. byte-jából beolvassa a rekord hosszúságát.
740–880	Programfile esetén meghatározza a file kezdőcímét az első adatblokkból, és az AA-ban tárolja.
890–980	Kiszámítja a lemez szabad blokkjainak számát úgy, hogy mindig a sávra vonatko-

zó BAM-kivonat első byte-ját olvassa és ezeket a BF-ben összegzi. A foglalt blokkok meghatározása BB = 644 – BF-fel történik.

990–1020 Relatív file esetén a rekordhosszúság (RL) és a file által lefoglalt blokkok számának segítségével kiszámítja az oldalszektorok (BS) és a rekordok (RC) számát. Mivel a DOS a relatív file minden 120 blokkjához hoz létre egy oldalszektort, a program a BS = BL/121 alapján az oldalszektorok számát határozza meg és ezt kerekíti fel a következő egész számra, majd a fennmaradó blokkokat 254-gyel beszorozva és a rekordhosszúsággal elosztva, megadja a file-ban lévő rekordok számát.

1040–1230 Az eredményeket kiírja képernyőre vagy nyomtatóra. (A file-paramétereket a REVERSE módban írja.)

1240–1280 További file-paraméter meghatározása.

Bár a program CBM 64-en készült, különösebb változtatások nélkül futtatható a VC 20-on is. Csak a 110. sorban kell változtatni a képernyő színének beállítására vonatkozó utasításban.

P. 45

```
100 clr
110 poke53280,0:poke53281,0:print"■"
120 printtab(6);"============================"
130 printtab(6);"file parameter megallapitas"
140 printtab(6);"============================"
150 print:print
160 print"Ez a program tajekoztatast ad a"
170 print"kivalasztott file parametereirol"
180 print"a kepernyore, vagy a nyomtatora."
190 :
200 print:print
210 print"filenev listat ker ? (i/n)"
220 getx$:ifx$<>"i"andx$<>"n"then220
230 ifx$="i"thenfl=1:gosub280
240 fl=0
250 input"a file neve";f$
260 iflen(f$)<=16 then 280
270 print"a filenev hosszu !!!":goto250
280 open15,8,15,"i0":open2,8,2,"#"
290 t=18:s=1
300 print#15,"b-r";2;0;t;s
310 print#15,"b-p";2;0
320 get#2,x$:ifx$=""thenx$=chr$(0)
325 t=asc(x$)
330 get#2,x$:ifx$=""thenx$=chr$(0)
340 s=asc(x$)
350 forx=0to7
360 print#15,"b-p";2;x*32+5
370 ff$=""
380 fory=0to15
390 get#2,x$:ifx$=""thenx$=chr$(0)
400 ifasc(x$)=160then430
410 ff$=ff$+x$
420 nexty
430 iff$=ff$then490
440 ifflthenprintff$
450 nextx
460 ift=0then480
470 goto300
480 close2:close15
485 iffl=0then print"filenev nem letezik!!!":goto210
490 ifflthenreturn
```

```
500 print#15,"b-p";2;x*32+2
510 get#2,x$:ifx$=""thenx$=chr$(0)
520 ty=asc(x$)
530 ft=tyand15
540 ifft=0thenft$="torolve van"
550 ifft=1thenft$="sequencialis"
560 ifft=2thenft$="program"
570 ifft=3thenft$="user"
580 ifft=4thenft$="relativ"
590 ifft>4thenft$="ismeretlen filetipus!!!":goto200
600 iftyand128thenge$="igen":goto620
610 ge$="nem"
620 iftyand64thensa$="igen":goto640
630 sa$="nem"
640 print#15,"b-p";2;x*32+30
650 get#2,x$:ifx$=""thenx$=chr$(0)
660 lb=asc(x$)
670 get#2,x$:ifx$=""thenx$=chr$(0)
680 hb=asc(x$)*256
690 bl=lb+hb
700 ifft<>4then740
710 print#15,"b-p";2;x*32+23
720 get#2,x$:ifx$=""thenx$=chr$(0)
730 rl=asc(x$)
740 ifft<>2then890
750 print#15,"b-p";2;x*32+3
760 get#2,x$:ifx$=""thenx$=chr$(0)
770 dt=asc(x$)
780 get#2,x$:ifx$=""thenx$=chr$(0)
790 ds=asc(x$)
800 open3,8,3,"#"
810 print#15,"b-r";3;0;dt;ds
820 print#15,"b-p";3;2
830 get#3,x$:ifx$=""thenx$=chr$(0)
840 lb=asc(x$)
850 get#3,x$:ifx$=""thenx$=chr$(0)
860 hb=asc(x$)*256
870 aa=lb+hb
880 close3
890 print#15,"b-r";2;0;18;0
900 bf=0
910 fori=4to140step4
920 ifi=72then960
930 print#15,"b-p";2;i
940 get#2,x$:ifx$=""thenx$=chr$(0)
950 bf=asc(x$)+bf
960 next
970 :
980 bb=644-bf
990 ifft<>4then1040
1000 :
1010 bs=bl/121:ifbs<>int(bs)thenbs=int(bs)+1
1020 rc=int(((bl-bs)*254)/rl)
1030 :
1040 print"■";"kepernyore vagy a nyomtatora (k/n)"
1050 getx$:ifx$<>"k"andx$<>"n"then1050
1060 re$=chr$(18):ra$=chr$(146)
1070 ifx$="k"thenopen1,3:print#1,chr$(147)
1080 ifx$="n"thenopen1,4
1090 print#1,"file parameterek     ";re$;f$;ro$
1100 print#1,"----------------------------------------"
1110 print#1,"file tipus:          ";re$;ft$;ra$:print#1
1120 print#1,"file beolvashato     ";re$;ge$;ra$:print#1
1130 print#1,"file vedett  -       ";re$;sa$;ra$:print#1
1140 print#1,"elfoglalt blokk      ";re$;bl;ra$:print#1
1150 ifft<>4then1200
1160 print#1,"rekordhossz          ";re$;rl;ra$:print#1
```

```
1170 print#1,"side-sektor blokk      ";re$;b$;ra$:print#1
1180 print#1,"adatblokk              ";re$;bl-bs;ra$:print#1
1190 print#1,"rekordok               ";re$;rc;ra$:print#1
1200 ifft=2thenprint#1,"startcim       ";re$;aa;ra$:print#1
1210 print#1,"szabad blokkok         ";re$;bf;ra$:print#1
1220 print#1,"foglalt blokkok        ";re$;bb;ra$:print#1
1230 close1
1240 print"folytassam  (i/n) ?"
1250 getx$:ifx$<>"i"andx$<>"n"then 1250
1260 close2:close15
1270 ifx$="i"then100
1280 end
```

4.1.2. File-védelem

A korábbiakban már utaltunk arra, hogy a tartalomjegyzékben bármely file védettnek minősíthető. A file-bejegyzés 0. byte-ja tartalmazza a file-típust. A 6. vagyis a 64 decimális értékű bit értéke jelöli, hogy a file védett-e vagy sem. Ha ez a bit 1, a file a SCRATCH utasítással nem törölhető. Miután azonban a DOS nem tartalmaz az említett bit beállítására szolgáló utasítást, így ehhez BASIC programot kell írni.

A P.46. jelzésű programmal:

- kiírhatjuk a lemezen tárolt file-ok nevét a képernyőre,
- védhetjük a file-okat,
- felszabadíthatjuk a file-okat,
- törölhetjük a file-okat.

Ezzel a programmal a védett file-ok is törölhetők. Védett file esetén a törlési kívánságot vissza kell igazolni.

A programot kimerítően dokumentáljuk, elképzelhető ugyanis, hogy az Olvasó a saját programjaihoz is tud belőle részeket használni.

A változók jegyzéke

GF	– A „file-nevek olvasása, keresése" rutinban beállítandó kapcsoló, amely jelzi, hogy a keresett file-név szerepel-e a lemezen vagy sem.
FL	– Kapcsoló, melynek értéke 1, ha a felhasználó listát kér a file-nevekről.
FT	– A file-típust tartalmazó változó
T	– A file-bejegyzés aktuális blokkjának sávja
S	– A file-bejegyzés aktuális blokkjának szektora
TT	– Az a sáv, amelyen a keresett file bejegyzése van
SS	– Az a szektor, amelyen a keresett file bejegyzése van
FF$	– A tartalomjegyzékből utolsóként olvasott file-név
F$	– A keresett file-név

A program dokumentálása

100	Képernyő színének beállítása	
110–230	Programfejléc és menű	
240–260	Elágazás választás szerint	
270	Vissza a menűhöz	
280–350	*A file-nevek listázása*	(rutin)

310	A képernyő törlése
320	Az FL kapcsoló beállítása listázáshoz
350	A kapcsoló visszaállítása, visszatérés a rutinból
360–600	*A file-ok védelme* (rutin)
390	Ugrás „A file-név beolvasása" rutinra
400	Ugrás „A file-nevek olvasása, keresése" rutinra
410–450	GF kapcsoló ellenőrzése (GF=0, ha a keresett file nincs a lemezen)
460–480	A file-típus beolvasása (FT)
490–500	A védettség ellenőrzése
510	A file védelme (6. bitbe 1.)
520–550	A file-típus átvitele a pufferbe és a blokk lemezre írása
560	A csatornák lezárása
570–600	A „file védett" üzenet kijelzése és visszatérés a rutinból
610–850	*Egy file védelmének feloldása* (rutin)
640	Ugrás „A file-név beolvasása" rutinra
650	Ugrás „A file-név olvasása, keresése" rutinra
660–700	GF kapcsoló ellenőrzése
710–730	A file-típus olvasása (FT)
740–750	Védettség ellenőrzése
760	A védelem feloldása (6. bit=0.)
770–800	A file-név átvitele a pufferbe és a blokk lemezre írása
810	A file lezárása
820–850	A rutin befejezése, visszatérés a menühöz
860–1170	*Egy file törlése* (rutin)
890	Ugrás „A file-név beolvasása" rutinra
900	Ugrás „A file-nevek olvasása, keresése" rutinra
910–950	GF kapcsoló ellenőrzése
960–980	A file-típus olvasása (FT)
990	Védettség ellenőrzése
1000–1030	Védett file-ra való utalás (a file azonban ennek ellenére törölhető)
1040–1060	Törlési kívánság visszaigazolása
1070	A 6. bit visszaállítása, ha a file védett volt
1080–1110	A file-típus átvitele a pufferbe és a blokk lemezre írása
1120	A lemezegység inicializálása
1130	A file törlése
1140–1170	A rutin befejezése
1190–1560	*A file-nevek olvasása, keresése* (rutin)
1220	A parancs- és adatcsatorna megnyitása
1230–1240	A tartalomjegyzék olvasása és a puffermutató beállítása
1250–1320	Írásvédettség ellenőrzése. A program a tartalomjegyzéket változtatás nélkül megpróbája felírni a lemezre (1250-es sor). Ha van írásvédelem a lemezen, a program 26, WRITE PROTECT ON hibaüzenetet küld.
1330	A sáv- és szektorváltozók kezdőértékeinek beállítása
1340–1350	A file-bejegyzés blokk olvasása és a puffermutató pozicionálása az első byte-ra
1400–1530	File-név olvasó ciklus. Ha FL=1, a program kiírja képernyőre a file-neveket, egyébként összehasonlítja az aktuális nevet a keresett névvel
1540–1560	Amennyiben a T változó 0-t tartalmaz, nincs további file-bejegyzés blokk, így a rutin véget ér

```
100 poke53280,0:poke53281,0:printchr$(158);chr$(14);
110 printtab(6);"============================"
120 printtab(6);"file torlese es vedelme."
130 printtab(6);"============================"
140 print:print
150 print"ezzel a programmal a file vedelemben
160 print"reszesitheto, felszabadithato es
170 print"torolheto."
180 print:print
190 printtab(6);"1 - a file-ok listazasa":print
200 printtab(6);"2 - egy file vedelme ":print
210 printtab(6);"3 - egy file felszabaditasa":print
220 printtab(6);"4 - egy file torlese":print
230 printtab(6);"5 - program befejezese":print
240 getx$:ifx$=""orval(x$)<1orval(x$)>5then240
250 ifval(x$)=5 then end
260 onval(x$)gosub280,360,610,860
270 goto 100
280 rem ------------------------------------
290 rem valamennyi filenev listazasa
300 rem ------------------------------------
310 printchr$(147):print"▮lista a filekrol▮"
320 fl=1:gosub1190
330 print:print"tovabb --->return"
340 inputx$
350 fl=0:return
360 rem ------------------------------------
370 rem        a file vedelme
380 rem ------------------------------------
390 gosub1580
400 gosub1190
410 ifgf=1then460
420 print"ilyen file nincs !!!":print
430 print"tovabb --->return"
440 inputx$:close2:close15
450 return
460 print#15,"b-p";2;x*32+2
470 get#2,x$:ifx$=""thenx$=chr$(0)
480 ft=asc(x$)
490 if(ft and 64)=0then510
500 print"a file mar vedett !!!":print:goto430
510 ft=(ft or 64)
520 print#15,"b-p";2;x*32+2
530 print#2,chr$(ft);
540 print#15,"b-p";2;0
550 print#15,"u2";2;0;tt;ss
560 close2:close15
570 print"a file vedve !!!"
580 print"tovabb --->return
590 inputx$
600 close2:close15:return
610 rem ------------------------------------
620 rem   a file-vedelem feloldasa
630 rem ------------------------------------
640 gosub1580
650 gosub1190
660 ifgf=1then710
670 print"ilyen file nincs !!!":print
680 print"tovabb ---->return"
690 inputx$:close2:close15
700 return
710 print#15,"b-p";2;x*32+2
720 get#2,x$:ifx$=""thenx$=chr$(0)
730 ft=asc(x$)
740 if(ft and 64)=64then760
```

```
750 print"ez a file nincs vedve !!!":print:goto680
760 ft=(ft and 255-64)
770 print#15,"b-p";2;x*32+2
780 print#2,chr$(ft);
790 print#15,"b-p";2;0
800 print#15,"u2";2;0;tt;ss
810 close2:close15
820 print"a filevedelem torolve !!!"
830 print"tovabb --->return"
840 inputx$
850 return
860 rem ------------------------------
870 rem      egy file torlese
880 rem ------------------------------
890 gosub1580
900 gosub1190
910 ifgf=1then960
920 print"ilyen file nincs !!!":print
930 print"tovabb --->return"
940 inputx$:close2:close15
950 return
960 print#15,"b-p";2;x*32+2
970 get#2,x$:ifx$=""thenx$=chr$(0)
980 ft=asc(x$)
990 if(ft and 64)=0 then1040
1000 print"figyelem !!!"
1005 print"________"
1010 print"a file vedett. toroljem ? (i/n)"
1020 getx$:ifx$=""orx$<>"i"andx$<>"n"then1020
1030 ifx$="n"then 1170
1040 print"biztos ? (i/n)"
1050 getx$:ifx$=""orx$<>"i"andx$<>"n"then1050
1060 ifx$="n"then 1170
1070 ft=(ft and 255-64)
1080 print#15,"b-p";2;x*32+2
1090 print#2,chr$(ft);
1100 print#15,"b-p";2;0
1110 print#15,"u2";2;0;tt;ss
1120 print#15,"i0"
1130 print#15,"s:"+f$
1140 print"a file torolve !!!"
1150 print"tovabb --->return"
1160 inputx$
1170 close2:close15:return
1180 :
1190 rem ------------------------------
1200 rem  file-ok olvasasa, keresese
1210 rem ------------------------------
1220 open15,8,15,"i0":open2,8,2,"#"
1230 print#15,"b-r";2;0;18;0
1240 print#15,"b-p";2;0
1250 print#15,"u2";2;0;18;0
1260 input#15,x1$
1270 ifval(x1$)<>26 then 1330
1280 print:
1290 print:
1300 print"tovabb --->return"
1310 inputx$
1320 close2:close15:return
1330 t=18:s=1:tt=18:ss=1
1340 print#15,"b-r";2;0;t;s
1350 print#15,"b-p";2;0
1360 get#2,x$:ifx$=""thenx$=chr$(0)
1370 t=asc(x$):ift<>0thentt=t
1380 get#2,x$:ifx$=""thenx$=chr$(0)
1390 s=asc(x$):ifs<>255thenss=s
1400 forx=0to7
```

```
1410 print#15,"b-p";2;x*32+2
1420 get#2,x$:ifx$=""thenx$=chr$(0)
1430 ifasc(x$)=0then1530
1440 print#15,"b-p";2;x*32+5
1450 ff$=""
1460 fory=0to15
1470 get#2,x$:ifx$=""thenx$=chr$(0)
1480 ifasc(x$)=160then1500
1490 ff$=ff$+x$
1500 nexty
1510 iffl thenprintff$:goto1530
1520 iff$=ff$thengf=1:goto1570
1530 nextx
1540 ift<>0then1340
1550 close2:close15
1560 iffl=0thenprint"ilyen file nincs !!!":fori=1to2000:next
1570 return
1580 rem ---------------------------------------
1590 rem           filenev megadasa
1600 rem ---------------------------------------
1610 print:print
1620 input"a file neve:";f$
1630 iflen(f$)<17then 1650
1640 print"a filenev tul hosszu (max 16 kar) !!!":print:goto1620
1650 gf=0:fl=0
1660 return
```

Ez a segédprogram is CBM 64-en készült és a VC 20-on is futtatható. Mindössze a 100. sort kell kihagyni, amely a CBM 64-nél a képernyő színét állítja be. Amennyiben különös súlyt kell fektetnünk egy optikailag kifogástalan képernyő-outputra, a 110–230. sorok összehangolhatók a VC 20-as képernyős ábrázolással.

4.1.3. A Backup program – a lemezek másolása

Kettős meghajtóegységgel ellátott alapgépeken a lemezek másolását olyan rendszerutasításokkal végezhetjük el, mint a BASIC 4.0 DUPLICATE vagy BACKUP utasításai. Az 1541-es egyszeres meghajtóegységen a lemezek másolására nem áll rendelkezésre BASIC utasítás, ezért a másolást az alapgépen keresztül felhasználói programokkal kell megvalósítani.
A másolási folyamat elvileg a következő: beolvassuk a másolandó lemez BAM-ját, nevét és ID-jét. A BAM-ból megállapítjuk, hogy a forráslemezen melyek a foglalt blokkok (fölösleges lenne ugyanis a szabad blokkok átmásolása). Ezután megnyitunk egy közvetlen elérésű file-t és annyi adatblokkot olvasunk be a tárba a forráslemezről, amennyi elfér (kb. 169 blokk). Most bekérünk egy lemezt (céllemez) és megformáljuk. A tárból előzetesen kiolvasott adatokat felírjuk az új lemezre. Ekkor ismét bekérjük a forráslemezt, ha még szükséges, és beolvassuk a következő 169 blokkot a tárba, majd kiírjuk a céllemezre. A teljes lemez átmásolás négy menetben zajlik le.
A közvetlen elérésű file olvasását és írását kivéve a program BASIC nyelven van írva. A gépi kódú program ugyanis jóval gyorsabb, mint a BASIC nyelven megírt 256 lépéses GET$ ciklus. A gépi kódú részek miatt ez a program csak Commodore 64-en futtatható. (A 16 K bővítésű VC 20-nál 11-szer kellene lemezt cserélni.)
Érdemes összevetni a kettős meghajtó másolási sebességét a VC 1541-esével. Az általunk közölt programnak egy teljes lemez átmásolásához kb. 20 percre van szüksége, a CBM 4040-es pedig ugyanezt kb. 3 perc alatt hajtja végre.
A program használata rendkívül egyszerű: az indítást követően a képernyőn megjelenített utasításoknak megfelelően a forráslemezt és a céllemezt kell cserélgetni.

P. 47.

```
100 rem *** backup ***
110 rem
120 poke56,23:clr:gosub640
130 open1,8,15:
140 dim b%(35,23),s%(35),z(7),a$(1)
150 a$(0)="masolat":a$(1)="eredeti":r=1
160 ad=23*256:gosub590
170 poke250,0:poke251,ad/256
180 gosub530:gosub290
190 printns" blokkot masolok.":print
200 t=1:s=0
210 fori=1to4:tt=4:ss=s:r=1:ifi=1then240
220 ifr=0andi=1thengosub450:goto240
230 gosub590
240 poke251,ad/256:forj=1to169
250 ifb%(t,s)=0thengosub570
260 s=s+1:ifs=s%(t)thent=t+1:s=0:ift=36thenj=169
270 next:ifrthenr=0:t=tt:s=ss:goto220
280 next:goto510
290 t=18:s=0:gosub570
300 ns=0:fort=1to35:s=0
310 ns=ns+s%(t)-peek(ad+4*t)
320 forj=1to3
330 b=peek(ad+4*t+j)
340 fori=0to7
350 b%(t,s)=b and z(i):s=s+1
360 nexti,j
370 fors=s%(t)to23
380 b%(t,s)=-1:nexts,t
390 fori=0to15
400 a=peek(ad+144+i)
410 ifa<>13 thenn$=n$+chr$(a)
420 next
430 i$=chr$(peek(ad+162))+chr$(peek(ad+163))
440 printn$,i$:return
450 print"Helyezzen be egy uj lemezt a meghajtoba"
460 print"es nyomja meg a [RETURN]-t.":print:poke198,0:close2
470 geta$:ifa$<>chr$(13)then470
480 print#1,"n0:"n$","i$
490 input#1,e1,e2$,e3,e4:if e1 then print"█drive:";e1;e2$;e3;e4"█":end
500 goto630
510 close2:close1:end
520 rem szektor/sav
530 fort=1to35
540 s%(t)=21:ift>17thens%(t)=19:ift>24thens%(t)=18:ift>30thens%(t)=17
550 next
560 fori=0to7:z(i)=2^i:next:return
570 ifrthenprint#1,"u1:"2;0;t;s:sysin:return
580 print#1,"b-p:"2;0:sysout:print#1,"u2:";0;t;s:return
590 close2:print"Kerem a "a$(r)"disklemezt a meghajtoba"
600 print"es nyomja meg a [RETURN]-t!":print:poke198,0
610 geta$:ifa$<>chr$(13)then610
620 open1,8,15:print#1,"i0"
630 open2,8,2,"#":return
640 fori=828 to 873:rem gepi rutin toltese
650 readx:pokei,x:s=s+x:next
660 data162,2,32,198,255,160,0,32,207,255,145,250
670 data200,208,248,230,251,32,204,255,96,198,1,162
680 data2,32,201,255,160,0,177,250,32,210,255,200
690 data208,248,230,251,32,204,255,230,1,96
700 ifs<>7312then print"hiba a data-sorokban levo adatoknal!": end
710 in=828:out=849:return
```

4.1.4. Az egyes file-ok átmásolása másik lemezre

Az alábbi programmal egyedi file-okat másolhatunk át egyik lemezről a másikra. Ez vonatkozik minden file-típusra, a relatív file-ok kivételével. A relatív file-okat olyan BASIC programmal másolhatjuk, amely a rekordokat egyenként beolvassa egy füzérbe, majd felírja az új file-ba.
Az alábbi program a forráslemezről beolvassa a másolandó file-t a tárba, majd a lemezcsere után a tárbeli adathalmazt folyamatosan felírja a céllemezen megnyitott azonos nevű file-ba.
Az adattároláshoz a gépben 39 kbyte áll rendelkezésre, ezzel a programmal tehát max. 166 blokkot tartalmazó file-okat tudunk másolni.
Hogy a másolás gyorsabb legyen, az adatok beolvasását és visszaírását egy kis gépi program végzi, amelyet DATA sorokban helyeztünk el.
A program kezelése nagyon egyszerű, futtatás közben csak az általa adott utasításokat kell betartani, azaz először a forrás-, majd a céllemezt kell az egységbe behelyezni.

P. 48.

```
100 print"█":poke56,12:clr
110 gosub 280
120 input"██file nev:";n$
130 print"█file tipus (s,p,u):";
140 gett$:ift$<>"s"andt$<>"p"andt$<>"u"then140
150 printt$:print
160 print"kerem az eredeti lemezt!"
170 print"[RETURN]-ra indul !!!":print
180 geta$:ifa$<>chr$(13)then 180
190 open2,8,2,"0:"+n$+","+t$
200 poke3,0:poke4,12:sys866
210 close2
220 print"kerem a cel lemezt!"
230 print"[RETURN]-ra indul !!!":print
240 geta$:ifa$<>chr$(13)then240
250 open2,8,2,"0:"+n$+","+t$+",w"
260 poke3,0:poke4,12:sys828
270 close2:end
280 fori=828to898
290 readx:pokei,x:s=s+x:next
300 data162,2,32,201,255,198,1,160,0,56,165,3
310 data229,5,165,4,229,6,176,13,177,3,32,210
320 data255,230,3,208,236,230,4,208,232,230,1,76
330 data204,255,162,2,32,198,255,160,0,32,207,255
340 data145,3,230,3,208,2,230,4,36,144,80,241
350 data165,3,133,5,165,4,133,6,76,204,255
360 ifs<>8634 thenprint"hiba a data-sorokban levo adatoknal!!!":end
370 return
```

4.1.5. A tartalomjegyzék beolvasása BASIC programból

Gyakran előfordul, hogy egy BASIC program a felhasználótól bekéri a feldolgozandó file nevét. Ha ezt a felhasználó elfelejtette, fel kell függesztenie a program futását, be kell töltenie a tartalomjegyzéket, a file kiválasztása után újra be kell töltenie a programot, majd azt újra el kell indítania. Ez az eljárás kissé nehézkes. Sokkal elegánsabb megoldás, ha a tartalomjegyzéket a program a felhasználó kívánságára futás közben betölti és kiírja a képernyőre. Erre a célra készítettük az alábbi programrészletet, amely rutinként beépíthető bármely adatfeldolgozó programba:

P. 49.

```
100 printchr$(147);
110 open15,8,15:open2,8,2,"#"
120 t=18:s=1
130 print#15,"b-r";2;0;t;s
140 print#15,"b-p";2;0
150 get#2,x$:ifx$=""thenx$=chr$(0)
160 t=asc(x$)
170 get#2,x$:ifx$=""thenx$=chr$(0)
180 s=asc(x$)
190 forx=0to7
200 print#15,"b-p";2;x*35+5
210 ff$=""
220 forry=0to15
230 get#2,x$:ifx$=""thenx$=chr$(0)
240 ifasc(x$)=chr$(160)then270
250 ff$=ff$+x$
260 nextry
270 ifa=0thena=1:printff$;:goto290
280 a=0:printtab(20);ff$
290 nextx
300 ift<>0then130
310 close2:close15
320 print"Nyomja le a [RETURN]-t!"
330 getx$:ifx$<>chr$(13)then330
340 end: rem ha ez egy szubrutin, akkor end helyett return
```

A program beolvassa a file-neveket a lemezről, majd képernyőre írja őket. Ha az eljárást szubrutinként használjuk, ami a GOSUB-bal hívható, az END utasítás helyett a 340. sorba a RETURN utasítást kell beírni.

4.2. A TEST/DEMO lemez segédprogramjai

A könyv bevezető fejezeteiben (1.1.2.) ígéretet tettünk arra, hogy a demonstrációs lemezen levő programokat részletesen bemutatjuk. Az az Olvasó, aki idáig eljutott a könyv tanulmányozásában, egészen biztosan hasznát veszi ezeknek a segédprogramoknak.

4.2.1. A DOS 5.1

A DOS 5.1 leegyszerűsíti a VC 1541-es DOS kezelését. A VC 20-as és a COMMODORE 64-es gépekhez egyaránt használható. A DOS 5.1 betöltése a VC 20-as gépen:

```
LOAD"VIC-20 WEDGE",8
RUN
```

Ez a DOS 5.1. töltőprogramja a VC 20-hoz. Ha a DOS 5.1-et a COMMODORE 64-en akarjuk használni, ez az alábbi utasításokkal történhet:

```
LOAD"C-64 WEDGE",8
RUN
```

Nézzük meg mit kínál a DOS 5.1?
A leggyakrabban alkalmazott utasításokat szimbólumokkal rövidíthetjük. Ha például a tarta-

lomjegyzéket akarjuk kiírni a képernyőre, az @$ vagy a >$ DOS 5.1 utasítással helyettesíthetjük a LOAD+LIST utasításokat.
A rövidített parancsok azzal az előnnyel is járnak, hogy nem írják felül a tárbeli programot.

A DOS 5.1 utasításai

ÍRÁSMÓD	FUNKCIÓ
@$ vagy >$	a tartalomjegyzék kiírása
@V vagy >V	a VALIDATE funkció
@C:... vagy >C:...	file-másolás (COPY)
↑file vagy /file	a programok betöltése
@ vagy >	a hibacsatorna lekérdezése és kijelzése
@N:... vagy >N:...	lemezformálás
@I vagy >I	a lemez inicializálása
@R:... vagy >R:...	egy file nevének megváltoztatása (RENAME)
@S:... vagy >S:...	egy file törlése (SCRATCH)

4.2.2. A COPY/ALL

A COPY/ALL program egy kétlemezes másoló. Két különböző egységszámú meghajtót feltételez, tehát ha esetleg két 8-as egységünk van, a DISK ADDR CHANGE programmal az egyiket át kell címezni. A program indítása után az alábbi üzenet jelenik meg a képernyőn:

```
disk copy all                jim butterfield
from unit? 8
```

Válaszul azt az egységszámot kell megadni, amelyben a forráslemez van. Ha ez a 8-as egység, csak a RETURN billentyűt kell megnyomni. A program ekkor bekéri az egység meghajtószámát (egyszeres meghajtóknál mindig nulla). Ugyanígy kell beállítani a másik egység- és meghajtószámot is.
Ha ez megtörtént, a program a következő kérdést teszi fel:

```
want to new the output disk
?n
```

Ez a kérdés azt jelenti, hogy szükség van-e a lemez formálására vagy sem. Következő lépésként a jokerrel (*) kiválaszthatjuk a másolandó file-okat. Ha az összes file-t másolni akarjuk, mindenütt csillagot kell beírni.
A program következő üzenete:

```
hold down 'y' or 'n' key to select
```

ami azt jelenti, hogy az Y vagy az N billentyűkkel dönthetünk, hogy egy file átkerüljön-e a céllemezre vagy sem.
Ha a másolási folyamat közben a file-ok mögött csillagok (***) jelennek meg, akkor sajnos a másolási folyamat nem volt hibamentes.
Amennyiben az összes file nem fér el a céllemezen, vagy a "*** output disk full" (a lemez

tele van), vagy a "do you have a new one" (van másik lemezed?) üzenet jelenik meg. Ha a válasz Y, úgy beadásával a fennmaradó file-ok egy másik, már megformált lemezre kerülnek. A másolási folyamat befejeztével a program kiírja a lemez szabad blokkjainak számát.

4.2.3. A DISK ADDR CHANGE program

Ezzel a programmal átcímezhetjük az egységeinket (4–15). A program indítása után a szükséges meghajtóegység kivételével az összes többit kapcsoljuk ki. Ezután először az eredeti, majd az új egységszámot kell megadni. A program lefutása után ismét bekapcsolhatjuk az összes többi meghajtóegységet.
A program az alábbi típusú meghajtóegységek átcímzésére alkalmas:

2031 DOS V2.6
2040 DOS V1.1
4040 DOS V2.1
4040 DOS V2.7
8050 DOS V2.5
8050 DOS V2.7
8250 DOS V2.7

4.2.4. A DIR

Ez egy kis segédprogram, amely az alábbi lehetőségeket biztosítja:

d – a tartalomjegyzéket kiírja
> – ezzel a karakterrel rövidíthetjük a lemezutasításokat (pl: > N: TEST, KN – egy lemez formálása)
q – kilépés a programból
s – a hibacsatorna lekérdezése.

4.2.5. A VIEW BAM

Ezzel a segédprogrammal a lemez blokkjainak foglaltsági térképét kérhetjük a képernyőre. A táblázat függőleges irányban a szektorokat, vízszintes irányban pedig a sávokat mutatja. A normális keresztek a szabad, a fordított keresztek a foglalt blokkokra vonatkoznak. Az n/a-val jelölt blokkok nem léteznek a sávon. A táblázat kiírása után a program kiírja a lemez nevét és a szabad blokkok számát.

4.2.6. A CHECK DISK

Ez a segédprogram írással és olvasással ellenőrzi a lemez blokkjait. Az aktuális blokk és az ellenőrzött blokkok száma megjelenik a képernyőn.

4.2.7. A DISPLAY T & S

Ha érdekel bennünket a lemezen lévő blokkok felépítése, s ezt ki akarjuk íratni képernyőre vagy nyomtatóra, ezt a programot használjuk. A program indítása után bekéri a kívánt sávot és szektort. A fenti célra kényelmesebb a könyvben közölt Disk-Monitor, miután azzal a blokkok tartalmának módosítására és visszaírására is van lehetőség.

4.2.8. A PERFORMANCE TEST

Ezzel a programmal a VC 1541-es meghajtóegység mechanikáját tesztelhetjük. A program az összes lemezműveletet elvégzi az egységen az alábbi sorrendben:

1. a lemez formálása
2. egy file megnyitása írásra
3. az adatok beírása a file-ba
4. a file lezárása
5. ugyanennek a file-nak megnyitása olvasásra
6. az adatok olvasása
7. a file lezárása
8. a file törlése
9. a 35-ös sáv beírása
10. az 1. sáv beírása
11. a 35-ös sáv olvasása
12. az 1. sáv olvasása

Minden lemezművelet elvégzése után megjelenik a hibacsatorna tartalma. Így megállapíthatjuk, hogy volt-e hibás művelet. A teszthez célszerű érdektelen adatokat tartalmazó lemezt használni, mert a tárolt adatok a program futása közben elvesznek.

4.3. BASIC bővítések és programok a VC 1541-es készülék kényelmes használatához

4.3.1. Tetszőleges hosszúságú füzérek beolvasása lemezről

A lemezes INPUT# utasításnak van egy nagy hátránya: a Commodore 64-en és a VC 20-on nincs lehetőség olyan adatok beolvasására, amelyek 88-nál több karaktert tartalmaznak. Ennek az az oka, hogy a gép INPUT pufferje csak ennyi karaktert tud tárolni. Ráadásul, ha egy füzéren belül vessző, vagy kettőspont van, azt a gép elválasztó karakternek tekinti és így a füzér fennmaradó része az INPUT lista következő változójába kerül. Ha az INPUT# utasítás csak egy 88 karakternél kisebb változót tartalmaz, a fennmaradó rész figyelmen kívül marad és csak a következő INPUT# utasítás (a következő RETURN) olvassa be.
Ha INPUT# helyett GET# utasítást használunk, ezek a problémák elkerülhetők, de a beolvasás nagyon lassú lesz.
Az alábbi kis gépi rutinban az INPUT# utasítást megváltoztatjuk, mégpedig úgy, hogy járulékos paraméterként az olvasandó karakterek számát is meg lehessen adni. A normális

INPUT# utasítástól való megkülönböztetésül ezt az utasítást INPUT*-nak nevezzük, amelynek a formája a következő:

INPUT* lf, len, var

Itt az lf az előzőleg megnyitott file logikai száma, a len adja a beolvasandó karakterek számát, a var a változó, amelybe a beolvasás történik.
A rutint a következő formában hívhatjuk:

```
100 OPEN 2,8,2, "ADAT"
110 INPUT* 2.100,A$
```

Ezzel egy 100 karakter hosszúságú füzért olvasunk be az A$-ba a megnyitott file-ból. Az eljárás különösen relatív file-ok esetén hasznos, ahol így pozicionálás után egy utasítással beolvasható a teljes rekord. A rekordot beolvasás után a MID$ utasítással kell mezőkre bontani (a 4.3.2. alfejezetben fogunk bemutatni egy egyszerű módszert a mezők előállítására). Az eljárás másik előnye, hogy a rekordokat nem kell lezárnunk RETURN-nel, tehát – relatív file-oknál – a teljes rekordhosszt használhatjuk:

```
100 OPEN 1,8,15
110 OPEN 2,8,2, "RELATÍV FILE,L," + CHR$(20)
120 PRINT# 1,"P" + CHR$(10) + CHR$(0) + CHR$(1)
130 PRINT# 2,"12345678901234567890";
140 PRINT# 1,"P" + CHR$(10) + CHR$(0) + CHR$(1)
150 INPUT* 2,20,A$
160 PRINT A$

12345678901234567890
```

Az alábbiakban közöljük a kazetta-pufferben tárolt gépi kódú program assembler-listáját és a gépi kódú program Commodore 64-en és VC 20-on használható BASIC töltőprogramját.

P.50

```
0000                 ;
0000                 ;
0000                 ;
0000                 ;input* lf,hossz,a$
0000                 input  = $85
0000                 stern  = $ac
0000                 basvec = $308
0000                 chrget = $73
0000                 chrgot = chrget+6
0000                 ;
0000                 ;                          c64 - verzio
0000                 ;
0000                 chkin  = $e11e
0000                 basin  = $e112
0000                 chkcom = $aefd
0000                 inter  = $a7ae
0000                 exeold = $a7e7
0000                 inpold = $abbf
0000                 fndvar = $b08b
0000                 strres = $b475
0000                 frestr = $b6a3
0000                 getbyt = $b79e
```

```
0000            ;
0000            varadr = $49
0000            clrch  = $ffcc
0000            para   = $61
0000            ;
0000            *      = 828
033c  a9 47     init   lda #<test
033e  a0 03            ldy #>test
0340  8d 08 03         sta basvec
0343  8c 09 03         sty basvec+1
0346  60               rts
0347            ;
0347  20 73 00  test   jsr chrget
034a  c9 85            cmp #input
034c  f0 06            beq found
034e  20 79 00         jsr chrgot
0351  4c e7 a7         jmp exeold          ; vissza a rutinra
0354  20 73 00  found  jsr chrget
0357  c9 ac            cmp #stern          ; uj input rutin
0359  f0 06            beq okstrn
035b  20 bf ab         jsr inpold
035e  4c ae a7         jmp inter
0361  20 9b b7  okstrn jsr getbyt-3
0364  20 1e e1         jsr chkin
0367  20 fd ae         jsr chkcom
036a  20 9e b7         jsr getbyt          ; hossz
036d  8a               txa
036e  48               pha                 ; tarolni
036f  20 fd ae         jsr chkcom
0372  20 8b b0         jsr fndvar          ; valtozot keresni
0375  85 49            sta varadr
0377  84 4a            sty varadr+1
0379  20 a3 b6         jsr frestr

line# loc   code        line

037c  68               pla                 ; hossz
037d  20 75 b4         jsr strres          ; teruletfoglalas a stingnek
0380  a0 02            ldy #2
0382  b9 61 00  store  lda para,y
0385  91 49            sta (varadr),y
0387  88               dey
0388  10 f8            bpl store
038a  c8               iny                 ; y=0
038b  20 12 e1  fetch  jsr basin
038e  91 62            sta (para+1),y
0390  c8               iny
0391  c4 61            cpy para
0393  d0 f6            bne fetch
0395  20 cc ff         jsr clrch
0398  4c ae a7         jmp inter           ; vissza az interpreter rutinba
039b                   ;

errors = 00000

symbol table

symbol value
 basin    e112   basvec   0308   chkcom   aefd   chkin    e11e
 chrget   0073   chrgot   0079   clrch    ffcc   exeold   a7e7
 fetch    038b   fndvar   b08b   found    0354   frestr   b6a3
 getbyt   b79e   init     033c   inpold   abbf   input    0085
 inter    a7ae   okstrn   0361   para     0061   stern    00ac
 store    0382   strres   b475   test     0347   varadr   0049
```

```
100 fori=828to922
110 readx:pokei,x:s=s+x:next
120 data 169,71,160,3,141,8,3,140,9,3,96,32,115,0,201,133
130 data 240,6,32,121,0,76,231,167,32,115,0,201,172,240,6,32
140 data 191,171,76,174,167,32,155,183,32,30,225,32,253,174,32,158
150 data 183,138,72,32,253,174,32,139,176,133,73,132,74,32,163,182
160 data 104,32,117,180,160,2,185,97,0,145,73,136,16,248,200,32
170 data 18,225,145,98,200,196,97,208,246,32,204,255,76,174,167
180 if s<>11096 thenprint"hiba a data-sorokban levo adatoknal!:end
190 sys828:print "rendben!"
```

```
100 for i =828to922
110 readq:pokei,q:s=s+q:next
120 data 169,71,160,3,141,8,3,140,9,3,96,32,115,0,201,133
130 data 240,6,32,121,0,76,231,199,32,115,0,201,172,240,6,32
140 data 191,203,76,174,199,32,155,215,32,27,225,32,253,206,32,158
150 data 215,138,72,32,253,206,32,139,208,133,73,132,74,32,163,214
160 data 104,32,117,212,160,2,185,97,0,145,73,136,16,248,200,32
170 data 15,225,145,98,200,196,97,208,246,32,204,255,76,174,199
180 ifs<>11442 thenprint"hiba a data-sorokban levo adatoknal!":end
190 sys828:print"rendben!"
```

4.3.2. Rekordok kényelmes előkészítése

A korábbi fejezetekben részletesen ismertettük a relatív file-ok szerkezetét. Tudjuk tehát, hogy a relatív file rögzített hosssúságú rekordokkal dolgozik. A rekord többnyire több mezőre oszlik, amelyek a rekordon belül fix pozíción kezdődnek és előre meghatározott hosszúsággal rendelkeznek.
Ha például egy új rekordot írunk a file-ba, többnyire külön inputokat használunk az egyes mezőkhöz. Mielőtt a teljes rekord írására kerülne sor, gondoskodnunk kell a rekord összeállításáról. Az egyes mezők hosszúságát ellenőrizni kell. Ha egy mező hosszabb az előre megadottnál, a fennmaradó részt "le kell vágni". Rövidebb mezők esetén a fennmaradó részt üres karakterekkel kell feltölteni.
Az alábbiakban két új BASIC utasítással ismerkedhetünk meg, amelyek a fenti feladatok megoldására kiválóan alkalmasak. Ezek az utasítások, melyeket gépi nyelven írtunk meg, tulajdonképpen BASIC bővítések, amelyeket SYS utasítással kell inicializálni ahhoz, hogy a későbbiekben hívhatók legyenek.
Az első utasítás az !STR$, amely megadható hosszúságú és megadható karakterből álló füzért állít elő. Pl. az

```
A$=!STR$(100," ")
```

utasítás egy 100 üresjeles füzért állít elő az A$ változóban.
A következő utasítás arra szolgál, hogy az adatmezőinket közvetlenül beírhassuk a fentiek szerint előállított füzérbe. Ha például a családnevet tartalmazó N$ változót az első pozíciótól számítva 25 karakter hosszan akarjuk elhelyezni az A$ füzérbe, utasításunk az alábbi formát ölti:

```
MID$ (A$,1,25,)=N$
```

Itt a MID$ utasítást ún. pszeudo-változóként az egyenlőségjel bal oldalán használjuk. Ekkor a következő történik:

Az N$ változó helyettesíti az A$ változó első 25 karakterét. Ha az N$ változó 25 karakternél hosszabb, a változó fennmaradó része figyelmen kívül marad. Akkor sincs baj, ha az N$ rövidebb, mint 25. A kitöltetlen helyeken maradnak az A$ eredeti karakterei (a mi esetünkben az üres jelek). Pontosan ez az, amit akartunk.
Az utasítások felhasználása egy BASIC programban:

```
200 INPUT "VEZETÉKNÉV"; N$
210 INPUT "KERESZTNÉV"; V$
220 INPUT "UTCA";       S$
230 INPUT "HÁZSZÁM";    NR$
240 INPUT "VÁROS";
250 INPUT "IR. SZÁM";
260 A$=!STR$ 94," "
270 MID$ (A$,1,25)      =N$
280 MID$ (A$,26,20)     =V$
290 MID$(A$,46,20)      =S$
300 MID$(A$,66,5)       =NR$
310 MID$ (A$,71,20)     =O$
320 MID$ (A$,91,4)      =P$
330 PRINT# 2, A$
```

P.51.

```
100 fori=51200 to 51479
110 read x:poke i,x: s=s+x :next
120 data169,13,160,200,141,10,3,140,11,3,76,107
130 data200,169,0,133,13,32,115,0,201,33,240,6
140 data32,121,0,76,141,174,32,115,0,201,196,240
150 data3,76,8,175,32,115,0,32,250,174,32,158
160 data183,138,72,32,253,174,32,158,173,36,13,48
170 data 12,32,170,177,165,100,208, 36,165,101, 76, 82
180 data200, 32,130,183,240,26,160,  0,177,34,133,  3
190 data104, 32,125,180,168,240, 7,165,3,136,145,98
200 data208,251,32,202,180,76,247,174,76,72,178,169
210 data118,160,200,141,8,3,140,9,3,96,32,115
220 data0,201,202,240,6,32,121,0,76,231,167,32
230 data 115,0,32,250,174,32,139,176,133,100,132,101
240 data133,73,132,74,32,163,182,160,0,177,100,72
250 data240,46,32,82,170,160,1,177,73,133,5,200
260 data177,73,133,6,32,253,174,32,158,183,138,240
270 data 23,202,134,4,32,121,0,201,41,208,4,169
280 data 255,208,12,32,253,174,32,158,183,138,208,3
290 data76,72,178,133,3,104,56,229,4,197,3,176
300 data 2,133,3,32,247,174,169,178,32,255,174,32
310 data158,173,32,163,182,160,2,177,100,133,81,136
320 data 177,100,133,80,136,177,100,240,211,197,3,176
330 data2,133,3,165,5,24,101,4,133,5,144,2
340 data230,6,164,3,136,177,80,145,5,192,0,208
350 data 247,76,174,167
360 ifs<>31128 then print"hiba a data-sorokban levo adatoknal!":end
370 sys51200:print"█ok.."
380 new
```

```
00001  0000          ;
00003  0000          ;
00004  0000                    *  =  $c800
00005  c800          chkauf  =  $aefa
00006  c800          chkzu   =  $aef7
```

```
c800                chkcom =  $aefd
c800                frmevl =  $ade9
c800                chkstr =  $ad8f
c800                frestr =  $ba63
c800                yfac   =  $b3a2
c800                chrget =  $73
c800                chrgot =  chrget+6
c800                getbyt =  $b79e
c800                integr =  $b1aa
c800                dscrpt =  $64
c800                stradr =  $62
c800                adr2   =  $fb
c800                adr1   =  $fb+2
c800                len1   =  3
c800                len2   =  4
c800                anzahl =  5
c800                start  =  6
c800                typflg =  12
c800                strcde =  $c4
c800                ilquan =  $b248
c800                syntax =  $af08
c800                pscode =  $b9
c800                vector =  $30a
c800                temp   =  len1
c800                ;
c800  a9 0d         lda    #<testin
c802  a0 c8         ldy    #>testin
c804  8d 0a 03      sta    vector
c807  8c 0b 03      sty    vector+1
c80a  4c 6b c8      jmp    midstr
c80d  a9 00         testin lda #0
c80f  85 0c         sta    typflg
c811  20 73 00      jsr    chrget
c814  c9 21         cmp    #33              ; ! van ?
c816  f0 06         beq    test2
c818  20 79 00      jsr    chrgot
c81b  4c 8d ae      jmp    $ae8d
c81e  4c 73 00      test2  jmp chrget
c821  c9 c4         cmp    #strcde
c823  f0 03         beq    string
c825  4c 08 af      jmp    syntax
c828                ;
c828                ; string$ - funkcio
c828                ;
c828  20 73 00      string jsr chrget
c82b  20 fa ae      jsr    chkauf           ; zarojel atugrasa
c82e  20 9e b7      jsr    getbyt
c831  8a            txa
c832  48            pha                     ; hosszt tarolni
c833  20 fd ae      jsr    chkcom
c836  20 e9 ad      jsr    frmevl
c839  24 0c         bit    typflg
c83b  30 0c         bmi    str              ; string
c83d  20 aa b1      jsr    integr
c840  a5 64         lda    dscrpt           ; felso byte
c842  d0 22         bne    ill              ; > 255
c844  a5 65         lda    dscrpt+1         ; also byte, hossz
c846  4c 50 c8      jmp    str2
c849  20 82 b7      str    jsr $b782        ; str beallitas, typflg numeriku
s
c84c  f0 18         beq    ill              ; hossza = 0
c84e  b1 22         lda    ($22),y          ; elso karakter
c850  85 03         str2   sta temp
c852  68            pla                     ; hosszt elohivni
c853  20 7d b4      jsr    $b47d            ; frestr
c856  a8            tay
c857  f0 07         beq    str3
```

```
c859  a5 03      lda     temp
c85b  88         loop1   dey
c85c  91 62      sta     (stradr),y       ; string eloallitas
c85e  d0 fb      bne     loop1
c860  20 ca b4   str3    jsr $b4ca; string verembe
c863  4c f7 ae   jmp     chkzu
c866  4c 48 b2   ill     jmp ilquan
c869                     ;
c869                     ;mid$(stringvaltozo,pozicio,hossz)=stringkifejezes
c869                     ;mid$(stringvaltozo,hossz)         =stringkifejezes
c869                     ;
c869                     midcod =  $ca
c869                     execut =  $308
c869                     execld =  $a7e7
c869                     varnam =  $45
c869                     varadr =  $49
c869                     dscrpt =  $64
c869                     tststr =  $ad8f
c869                     getvar =  $b08d
c869                     setstr =  $aa52
c869                     test   =  $aeff
c869                            *=  3
0003                     laenge *=  *+1
0004                     poston *=*+1
0005                     varstr *=*+2
0007                     gleich =  $b2
0007                     zeig2  =  $50
0007                            *=  $c86b
c86b                     ;
c86b  a9 76      midstr lda #<midtst
c86d  a0 c8      ldy     #>midtst
c86f  8d 08 03   sta     execut
c872  8c 09 03   sty     execut+1
c875  60         rts
c876  20 73 00   midtst jsr chrget
c879  c9 ca      cmp     #midcod          ; a kod mid$ ?
c87b  f0 06      beq     mid              ; igen
c87d  20 79 00   jsr     chrgot
c880  4c e7 a7   jmp     execld           ; vissza a szokasos funkcio vegr
ehajtasahoz
c883  20 73 00   mid     jsr chrget       ;kovetkezo karakter
c886  20 fa ae   jsr     chkauf           ; zarojel atugrasa
c889  20 8d b0   jsr     getvar           ; valtozo olvasasa
c88c  85 64      sta     dscrpt
c88e  84 65      sty     dscrpt+1
c890  85 49      sta     varadr
c892  84 4a      sty     varadr+1
c894  20 63 ba   jsr     frestr
c897  a0 00      ldy     #0
c899  b1 64      lda     (dscrpt),y
c89b  48         pha                      ; hossz tarolva
c89c  f0 2e      beq     ill1
c89e  20 52 aa   jsr     setstr           ; string a ram-ba
c8a1  a0 01      ldy     #1
c8a3  b1 49      lda     (varadr),y
c8a5  85 05      sta     varstr           ; valtozocim beallitas
c8a7  c8         iny
c8a8  b1 49      lda     (varadr),y
c8aa  85 06      sta     varstr+1
c8ac  20 fd ae   jsr     chkcom
c8af  20 9e b7   jsr     getbyt           ; pozicio olvasasa
c8b2  8a         txa
c8b3  f0 17      beq     ill1
c8b5  ca         dex
c8b6  86 04      stx     poston
c8b8  20 79 00   jsr     chrgot
c8bb  c9 29      cmp     #41              ; ) - kifejezes vege?
```

```
c8bd  d0 04        bne     next
c8bf  a9 ff        lda     #$ff                ; maximalis hossz
c8c1  d0 0c        bne     store
c8c3  20 fd ae     next    jsr chkcom
c8c6  20 9e b7     jsr     getbyt              ; hossz olvasasa
c8c9  8a           txa
c8ca  d0 03        bne     *+5
c8cc  4c 48 b2     ill1    jmp ilquan
c8cf  85 03        store   sta laenge
c8d1  68           pla
c8d2  38           sec
c8d3  e5 04        sbc     poston
c8d5  c5 03        cmp     laenge
c8d7  b0 02        bcs     ok
c8d9  85 03        sta     laenge
c8db  20 f7 ae     ok      jsr chkzu           ; zarojel
c8de  a9 b2        lda     #gleich
c8e0  20 ff ae     jsr     test
c8e3  20 e9 ad     jsr     frmevl              ; kifejezes olvasas
c8e6  20 63 ba     jsr     frestr
c8e9  a0 02        ldy     #2
c8eb  b1 64        lda     (dscrpt),y
c8ed  85 51        sta     zeig2+1
c8ef  88           dey
c8f0  b1 64        lda     (dscrpt),y
c8f2  85 50        sta     zeig2
c8f4  88           dey
c8f5  b1 64        lda     (dscrpt),y
c8f7  f0 d3        beq     ill1                ; hiba, ha nulla
c8f9  c5 03        cmp     laenge
c8fb  a5 05        ok1     lda varstr
c8fd  18           clc
c8fe  65 04        adc     poston
c900  85 05        sta     varstr
c902  90 02        bcc     *+4
c904  e6 06        inc     varstr+1
c906  a4 03        ldy     laenge
c908  88           loop    dey
c909  b1 50        lda     (zeig2),y           ;karakter a kifejezesbol
c90b  91 05        sta     (varstr),y; atvitel a string valtozoba
c90d  c0 00        cpy     #0
c90f  d0 f7        bne     loop
c911  4c ae a7     jmp     $a7ae               ; vissza az interpreter rutinba
c914               .end
```

4.3.3. A spooling – közvetlen nyomtatás lemezről

Ha alapgépünkhöz a lemezegységen kívül nyomtatót is csatlakoztatunk, kihasználhatjuk a soros busz speciális tulajdonságait. A file-okat közvetlenül átadhatjuk a lemeztárból a nyomtatóra anélkül, hogy ezt a folyamatot byte-ról byte-ra kellene végrehajtanunk a gép tárán keresztül. Ha egy tetszőleges szöveget soros file-ként tárolunk a lemezen és ki akarjuk nyomtatni, arra a következő program alkalmas:

P.52

```
100 open1,4
110 open2,8,2,"0:text":rem szoveg file
120 get#2,a$:ifst=64then140
130 print#1,a$;:goto120
140 close2:close1
150 end
```

A file végének felismeréséig a program beolvassa a karaktereket a lemezről és a nyomtatóra továbbítja. Ezután lezárjuk a két file-t és a program futása befejeződik. Ez egy szabályos adatátvitel, melynek végrehajtása BASIC programmal történik.

Hogyan működik a közvetlen nyomtatást végző program?
A program először megnyitja a két file-t. A nyomtatóhoz küld egy utasítást az adatok vételére (LISTEN), a lemezegységhez pedig az adatok továbbítására (TALK). Ettől kezdve a lemezegység automatikusan továbbítja az adatokat a nyomtatóra egészen addig, amíg file-véget nem érzékel. Közben használhatjuk a gépet anélkül, hogy ez az adatátvitelt zavarná (csak a soros buszra kapcsolt perifériák nem használhatók ez idő alatt).
A gyakorlatban a megoldás gépi kódú programmal történik. Amikor indítani akarjuk a spoolingot, a programot behívjuk és átadjuk neki a továbbítandó file nevét. A

SYS 828,"TEXT"

utasítás a lemezen lévő TEXT file-t kiírja nyomtatóra. Az átvitel megkezdése után a gép azonnal visszajelentkezik READY üzenettel. Biztosítandó, hogy a gép nincs lekötve átviteli célokra, kihúzhatjuk a lemezegységből kivezető buszkábelt és így a lemezegység már csak a nyomtatóval van összekapcsolva. A file-t, amely a nyomtatás befejeztével nyitva marad (az egységen a vörös színű LED világít), egy file-név nélküli SYS utasítással zárhatjuk le (természetesen ismét vissza kell dugni az egységhez vezető kábelt).
A SYS 828 utasítással a folyamatban lévő adatátvitelt be is fejezhetjük. Az alábbiakban közöljük a gépi programot, valamint a COMMODORE 64-hez és a VC 20-hoz alkalmazható betöltőprogramokat.

P.53.

```
033C  20 79 00   JSR   0079    ; CHRGOT ; van még karakter?
033F  F0 33      BEQ   0374    ; NINCS
0341  20 E7 FF   JSR   FFE7    ; CLALL  ; csatornák zárása
0344  20 54 E2   JSR   E254
0347  A6 B7      LDX   B7      ; filenév hossza
0349  F0 38      BEQ   0383
034B  A9 02      LDA   #02
034D  A2 08      LDX   #08
034F  A0 02      LDY   #02
0351  20 BA FF   JSR   FFBA    ; SETLFS
0354  20 C0 FF   JSR   FFC0    ; OPEN   ; logikai file nyitása
0357  A9 04      LDA   #04
0359  20 B1 FF   JSR   FFB1    ; LISTEN ; parancs az egységnek
035C  20 BE ED   JSR   EDBE
035F  A2 02      LDX   #02
0361  20 C6 FF   JSR   FFC6    ; CHKIN  ; csatorna nyitás INPUT-nak
0364  20 BE ED   JSR   EDBE
0367  20 85 EE   JSR   EE85
036A  20 97 EE   JSR   EE97
036D  A9 00      LDA   #00
036F  85 99      STA   99
0371  85 98      STA   98
0373  60         RTS
0374  A9 01      LDA   #01
0376  85 98      STA   98
0378  20 AE FF   JSR   FFAE    ; UNLISTEN ; parancs
037B  20 AB FF   JSR   FFAB    ; UNTALK   ; parancs
037E  A9 02      LDA   #02
0380  4C C3 FF   JMP   FFC3    ; CLOSE    ; logikai file zárása
0383  4C 08 AF   JMP   AF08    ; ERROR    ; hibaüzenet kiírása
```

```
100 for i =828to901:readq:pokei,q:s=s+q:next
110 data 32,121,0,240,51,32,231,255,32,84,226,166,183,240,56,169
120 data 2,162,8,160,2,32,186,255,32,192,255,169,4,32,177,255
130 data 32,190,237,162,2,32,198,255,32,190,237,32,133,238,32,151
140 data 238,169,0,133,153,133,152,96,169,1,133,152,32,174,255,32
150 data 171,255,169,2,76,195,255,76,8,175
160 ifs<>9598 then print"hiba a data-sorokban levo adatoknal":end
170 print"rendben"
```

```
100 for i =828to901
110 readq:pokei,q:s=s+q:next
120 data 32,121,0,240,51,32,231,255,32,81,226,166,183,240,56,169
130 data 2,162,8,160,2,32,186,255,32,192,255,169,4,32,177,255
140 data 32,197,238,162,2,32,198,255,32,197,238,32,132,239,32,160
150 data 228,169,0,133,153,133,152,96,169,1,133,152,32,174,255,32
160 data 171,255,169,2,76,195,255,76,8,207
170 if s<>9642 then print"hiba a datasorokban levo adatoknal":end
180 print"rendben"
```

4.4. Az overlay (átfedés) technika – a programok láncolása

Bonyolultabb adatfeldolgozó rendszerek kifejlesztésénél felmerülhet az a probléma, hogy nem tudjuk az összes adatkezelési funkciót egy programmal megoldani. Ennek több oka lehet – a "mindentudó" program túlságosan bonyolult, és emiatt meglehetősen lassú lenne, sőt az is előfordulhat, hogy egy mindenre felkészített program nem fér el a tárban. Célszerű tehát az ilyen rendszereket *menütechniká*val megoldani. Így a rendszer amellett, hogy minden adatkezelő funkciót képes ellátni, olyan önálló programokból épül fel, amelyek egy menüprogramból hívhatóak. Ezzel a megoldással a tárban mindig csak az éppen aktív programrész tartózkodik, és egy főprogram (menüprogram) szervezi a részprogramok hívását, láncolását.
Az ilyen rendszerek megvalósításánál a legnagyobb gondot a részprogramok tárbeli elhelyezése és a köztük szükséges információcsere megoldása okozza.
Hogyan lehet elérni, hogy az egyik program a másik változóinak értékeit használni tudja? Azok a megoldások, amelyeket mi az alábbiakban javasolunk, természetesen a CBM gépek társzervezésére épülnek, tehát elsősorban ezeken a gépeken használhatók.
Tegyük fel, hogy egy tárban lévő program betölt egy másikat. A 2. program csak akkor tudja átvenni az első változóit, ha hossza kisebb vagy egyenlő, mint az őt behívó (1. számú) program hossza. Ha ugyanis programból adjuk ki a LOAD utasítást, akkor a régi programra vonatkozó programvég mutatók megmaradnak és a vezérlés az új program első sorára adódik.
Nézzünk erre egy példát:

P.54.

```
100 rem program 1
110 rem ez a program hosszabb, mint a masodik
120 a=1000
130 load"p54/2",8: rem program 2
```

```
100 rem program 2
110 printa
```

Ha viszont az utántöltött program nagyobb az eredetinél, a változók egy része a tárban a második programmal felülíródik, tartalmuk határozatlan lesz, sőt a változókhoz történő értékhozzárendeléskor a programnak az első programon túlnyúló része is megrongálódhat.
A társzervezési problémákon kívül még egy CBM sajátosságra kell figyelnünk. Ha betartjuk a fenti szabályt – egymásratöltésnél a tárbeli program hossza nagyobb a betöltött programénál – akkor az eredeti program változói sértetlenül megmaradnak. Ennek magyarázata, hogy a változóterület a BASIC program után van. A szövegkonstansok azonban nem a változóterületen vannak, ugyanis a címtáblázatban a változónévhez rendelt mutató (pointer) arra a programrészre mutat, ahol a változó értéket kapott.
Tegyük fel, hogy az eredeti programban volt egy ilyen sor:

```
100 A$="SZOVEG"
```

A második program betöltésekor a 100-as sor nyilván megváltozik, de az A$ változóhoz rendelt tárcím marad. Az A$ tartalma tehát határozatlan.
Ezen a problémán úgy segíthetünk, hogy a szövegkonstansokat valamilyen mesterséges módon áthelyezzük a változóterületre, pl. végzünk velük egy műveletet, ami nem változtatja meg az értéküket:

```
100 A$="SZOVEG" +""                    (egy üres füzér hozzáadásával)
```

Ugyanez vonatkozik a függvénydefiníciókra, mert a mutató itt is a programterületen levő definícióra mutat.

```
100 DEF FN A(X) = 0,5 * EXP (-X*X)
```

Ha a betöltött programnak nincs szüksége a hívó (tárbeli) program változóira, az átfedési problémákat nagyon egyszerűen megoldhatjuk.
Mindössze annyit kell tenni, hogy közvetlenül betöltés után – a végcím ismeretében – a BASIC programvég mutatót az új programnak megfelelően, két POKE utasítással átállítjuk:

```
POKE 45, PEEK(174): POKE 46, PEEK (175): CLR
```

A CLR utasítás feltétlenül szükséges. Ennek a sornak elsőként kell elhelyezkednie az utántöltött programban. Így a változók átadása nélkül bármilyen nagyságú programokat hívhatunk egymás után. Egy másik, de korántsem ilyen elegáns lehetőség, hogy a töltőutasítást beírjuk a billentyűzet pufferébe, amely a 2. programot közvetlen üzemmódban automatikusan tölti. Kiírjuk a képernyőre a LOAD és a RUN utasítást és a billentyűzet-puffert, beírjuk a HOME és RETURN kódjait. Ezeket az utasításokat END-del kell befejezni. Az operációs rendszer ekkor közvetlen üzemmódban behozza a billentyűzet-puffer tartalmát, s így a LOAD és RUN utasítást végrehajtja, ami a program betöltését és futtatását eredményezi. Miután mindez közvetlen üzemmódban történik, a program végcíme automatikusan beállítódik, a változók törlődnek és a következő RUN utasítás a programot indítja.
Ennek a módszernek jelentős hátránya, hogy a képernyőre kiírt töltőutasítás az esetleges képernyőmaszkot elrontja.
A kivitelezés:

P.55.

```
1000 printchr$(147)"load"chr$(34)"program nev2"chr$(34)",8"
1010 print:print:print:print
1020 print"run"
1030 poke631,19:poke632,13:poke633,13
1040 poke634,13:poke635,13:poke636,13
1050 poke198,6:end
```

A gépi kódú programok utántöltése másféle technikával történik. Ha egy BASIC program gépi programokat használ, azokat többnyire indításkor betölti. Nem szabad megfeledkeznünk az alábbiakról: Mindig ügyelnünk kell arra, hogy a gépi kódú programokat abszolút jelleggel töltsük be, azaz meghatározott tárterületre. Ha egy program betöltésekor nem adunk meg másodlagos címet, az operációs rendszer feltételezi, hogy ez BASIC program, a betöltést a BASIC RAM kezdőcímétől kezdi (C 64-en 2049). A gépi kódú programok azonban csak akkor futtathatók, ha a tervezett címmel töltjük be őket, hiszen fix ugrási címeket tartalmaznak. Így csak a következő megoldás a helyes:

```
LOAD "GEPIPROGRAM",8,1
```

Arról sem szabad megfeledkeznünk, hogy program-üzemmódos betöltéskor a vezérlés mindig az első BASIC sorra adódik, és mivel az operációs rendszer minden LOAD után feltételezi, hogy BASIC programot töltött be, ez gépi kódú programok betöltésekor végtelen ciklushoz vezetne:

```
100 LOAD "GEPIPROGRAM",8,1
```

Ha most kihasználjuk, hogy utántöltéskor a változók megmaradnak, az alábbiak szerint kiküszöbölhetjük a végtelen ciklust:

```
100 IF A=0 THEN A=1 :LOAD "GEPIPROGRAM",8,1
110 ........
```

Ha a programot RUN-nal indítjuk, az A értéke még nulla, tehát a THEN ág hajtódik végre, ekkor az A értéke 1 lesz, és a gépi kódú program betöltődik. Betöltés után a program ismét az elejétől kezdve fut, de mivel A = 1 (értéke megmaradt!), a LOAD-ot nem hajtja végre, hanem áttér a következő utasításra.
Pontosan így járhatunk el, ha több gépi kódú programot kell egymás után betölteni:

```
100 IF A=0 THEN A=1 : LOAD "1. PROGRAM",8,1
110 IF A=1 THEN A=2 : LOAD "2. PROGRAM",8,1
120 IF A=2 THEN A=3 : LOAD "3. PROGRAM",8,1
```

4.5. A Merge – BASIC programok összefűzése

A programozási munkát gyakran megkönnyíti, ha az egyes részeket előre kidolgozzuk, majd a kész egységeket egy programmá fűzzük össze.
Eddigi ismereteink alapján ez nem lehetséges, hiszen ha a tárban van egy programrész, és

lemezről betöltünk egy másikat, a második rész felülírja az elsőt. Olyan lemezművelet pedig nincs, ami a két lemezen levő programot „összeillesztené".
A feladat megoldható, de ehhez tisztában kell lennünk a BASIC programok tárbeli elhelyezésével:

```
NL NH      a következő programsorra mutató pointer (alsó-, felső byte)
ZL ZH      programsorszám (alsó-, felső byte)
XX YY ZZ ...... sorok tartalma
00         a sor végének jelölése
NL NH      a következő programsorra mutató pointer (alsó-, felső byte)
ZL ZH      programsorszám
XX YY ZZ ...... sorok tartalma
00         a sor végének jelölése
```

A program végén még két nulla byte áll (a sorok végét jelző nulla byte-tal együtt három).
A program kezdő- és végcíme a tárban a 43-as, illetve 44-es (alsó, felső) és a 45-ös, illetve 46-os tárcímeken helyezkedik el.

```
PRINT PEEK (43) + 256*PEEK(44),
```

A fenti utasítással kiírhatjuk a BASIC program kezdőcímét (ez C–64-en 2049), az alábbival pedig a végcímet:

```
PRINT PEEK (45) + 256*PEEK (46),
```

Az utóbbi a három nulla byte utáni byte-ra mutat.
Kihasználva, hogy minden parancs üzemmódban kiadott LOAD a BASIC programot a 43-as és 44-es tárcímeken talált címtől kezdve tölti, ha ezeket a címeket egy, a már tárban levő program végcímére állítjuk, egy esetleges második program ettől a címtől kezdve töltődik be (azaz pontosan az első után).
Mindenekelőtt betöltjük az első programot a tárba, majd ezután beolvassuk a program végcímét:

```
A =PEEK(45) + 256*PEEK(46)
```

Ezt az értéket 2-vel csökkentjük, hogy a program végén a két nulla byte is átíródhassék,

```
A = A-2
```

Megjegyezzük az eredeti BASIC kezdőcímet.

```
PRINT PEEK(43), PEEK(44)
```

A fent kapott végcím lesz a BASIC kezdőcím.

```
POKE 43, A AND 255 : POKE 44, A/256
```

Betöltjük a második programot:

```
LOAD "2. PROGRAM",8
```

Visszaállítjuk az eredeti kezdőcímet (C–64-en 2049=256*8+1, azaz az alsó byte értéke 1, a felső byte-é 8).

```
POKE 43,1 : POKE 44,8
```

majd kilistázhatjuk a programot, amely most már mindkét részt tartalmazza, tehát egy programként összefűzve tárolhatjuk a lemezen.
Az eljárás során vigyáznunk kell arra, hogy a második program csak olyan sorszámokat tartalmazhat, amelyek nagyobbak az első program legnagyobb sorszámánál, ellenkező esetben ugyanis erre a sorszámra sohasem hivatkozhatunk GOTO-val, vagy GOSUB-bal.
A programok összefűzési lehetőségét ismerve kialakíthatunk egy szubrutingyűjteményt, amely a leggyakrabban használt eljárásokat tartalmazza lehetőleg olyan sorszámozással, amit egy tervezett főprogramban általában nem használunk.
További munkánk során az aktuális programhoz az éppen szükséges szubrutint egyszerűen hozzáfűzzük.

4.6. A Disk-Monitor

Az előző fejezetben már több szó esett erről a hasznos segédprogramról, amely C 64-en és VC 20-on egyaránt használható, és a haladó programozók elengedhetetlen munkaeszköze.

Leglényegesebb funkciói:
a lemez tartalmát – megjeleníthetjük képernyőn
– módosíthatjuk
– visszaírhatjuk lemezre.

Gyakorlott programozók tudják, hogy ezeket az egyszerű lehetőségeket milyen sokrétűen fel lehet használni a programozói munka során.

A program gépi kódban készült. Választható funkciók:
- egy blokk beolvasása a lemezről
- egy blokk kiírása a lemezre
- egy blokk kiírása a képernyőre
- egy blokk módosítása a képernyőn
- lemezkezelő utasítások továbbítása
- visszatérés a BASIC-re

A program indítása után bejelentkezik a:

```
DISK-MONITOR V1.0
>
```

majd várja utasításunkat.

1. @ – beolvassa az egység hibaüzenetét, és kiírja a képernyőre: 00, ok, 00, 00

2. Lemezműveletek
   Ha az @ jel után beírjuk valamelyik lemezkezelő utasítás kezdőbetűjét, a program végrehajtja a megfelelő műveletet.
   BASIC megfelelői:

   OPEN 15,8,15
   PRINT 15; „utasítás"
   CLOSE 15
   >@ I – inicializálja a lemezt
   >@ N: lemeznév, ID – megformálja a lemezt stb.

3. R és W – blokk írása-olvasása
   A program legfontosabb funkciója azonban a lemez minden egyes blokkjának közvetlen olvasása és írása. Erre szolgál az R és a W utasítás. Az R (READ) beolvas egy blokkot a lemezről, a W (WRITE) felír egy blokkot a lemezre.
   Mindkét esetben meg kell adnunk a kívánt blokk hexadecimális címét, azaz a sávot és a szektort hexadecimális alakban.
   Írjuk be a

   >R 12 01

   utasítást. A sáv és a szektor hexadecimális értékeit egy üres jellel választjuk el egymástól.

4. M – egy blokk tartalmának megjelenítése képernyőn
   Az M vezérlőutasítás előtt természetesen végre kell hajtani egy beolvasást az R utasítással.

```
DISK MONITOR

>M
>:00  12  04  82  11  01  47  52  41  .....GRA
>:08  46  49  4B  20  41  49  44  2E  FIK AID.
>:10  53  52  43  A0  A0  00  00  00  SRC ....
>:18  00  00  00  00  00  00  15  00  ........
>:20  00  00  82  13  00  48  50  4C  .....HPL
>:28  4F  54  2E  53  52  43  A0  A0  OT.SRC
>:30  A0  A0  A0  A0  A0  00  00  00      ....
>:38  00  00  00  00  00  00  05  00  ........
>:40  00  00  82  13  03  56  50  4C  .....VPL
>:48  4F  54  2E  53  52  43  A0  A0  OT.SRC
>:50  A0  A0  A0  A0  A0  00  00  00      .....
>:58  00  00  00  00  00  00  00  00  ........
>:60  00  00  82  13  00  4D  45  4D  .....MEM
>:68  2E  53  52  43  A0  A0  A0  A0  . SRC
>:70  A0  A0  A0  A0  A0  00  00  00  .    ....
>:78  00  00  00  00  00  00  06  00  ........
```

```
>:80  00  00  82  13  08  4D  45  4D  .....MEM
>:88  2E  4F  42  4A  A0  A0  A0  A0  . OBJ
>:90  A0  A0  A0  A0  A0  00  00  00
>:98  00  00  00  00  00  00  01  00  ........
>:A0  00  00  82  10  00  53  57  41  .....SWA
>:A8  50  2E  53  52  43  A0  A0  A0  P.SRC
>:B0  A0  A0  A0  A0  A0  00  00  00       ...
>:B8  00  00  00  00  00  00  04  00  ........
>:C0  00  00  82  10  01  4D  41  54  ....MAT
>:C8  52  49  58  2E  53  52  43  A0  RIX.SCR
>:D0  A0  A0  A0  A0  A0  00  00  00       ---
>:D8  00  00  00  00  00  00  0D  00  ........
>:E0  00  00  82  13  OC  47  41  55  .....GAU
>:E8  53  53  2E  54  45  53  54  AO  SS.TEST
>:F0  A0  A0  A0  A0  A0  00  00  00       ....
>:F8  00  00  00  00  00  00  01  00  .......
```

Vizsgáljuk meg közelebbről a fenti lemezkivonatot. A kettőspont utáni első hexadecimális szám adja meg a blokkon belül a következő 8 byte kezdősorszámát. 00 a blokkon belüli első byte sorszáma, mivel a számozás 00-tól FF-ig, azaz 0-tól 255-ig tart. Ezután 8 byte tartalma következik (VC 20-nál 4 byte-é). Ha a byte-ok tartalma nem kinyomtatható karakter ($00-tól $1F-ig és $80-tól $9F-ig terjedő ASCII kódok), a megfelelő helyen egy pont van. Ha ismét beütjük az M utasítást, a blokk második része jelenik meg (a teljes blokk ugyanis nem fér el egyszerre a képernyőn).
Ha egy blokknak csak egy meghatározott részét akarjuk látni, a kívánt rész határait paraméterként beírhatjuk az M utasításba.

A blokk első részének kiíratása:

```
>M 00 7F
```

A második rész kiíratása:

```
>M 80 FF
```

(VC 20-nál negyed blokk is megjeleníthető).

5. A blokk tartalmának módosítása
Ha bármilyen adatot meg akarunk változtatni, egyszerűen vigyük a kurzort a megfelelő helyre, írjuk át a megfelelő byte-ot, s nyomjuk le a RETURN-t. Hatására az adott byte-érték tartalmának új értéke jelenik meg, és egyidejűleg az ASCII karakter is megváltozik a jobb oldali részben.
Az így megváltoztatott blokkot a W utasítással visszaírhatjuk a lemezre. Természetesen a lemezcímet most is hexadecimálisan kell megadni.
A következő

```
>W 12 01
```

utasítás visszaírja a blokkot a 18-as sáv 1. szektorába, vagyis oda, ahonnan előzőleg a blokkot olvastuk.

6. Az X – visszatérés a BASIC-be

Hatására a gép ismét kiírja a READY-t. Ha később újra használni akarjuk a Disk-Monitort, nem kell újra betöltenünk, mert a Commodore 64-nél a SYS 49152-vel, a VC 20-nál pedig a SYS 6690-nel ismét meghívhatjuk.

Mielőtt egy lemez tartalmát módosítanánk, célszerű róla egy másolatot készíteni! Ha ugyanis egy blokk megváltoztatása vagy írása kapcsán bármilyen hibát vétünk, elveszíthetjük a lemezen eddig tárolt fontos információkat. Az is előfordulhat, hogy a lemez tartalmát a megszokott módon nem tudjuk többé visszaolvasni.

Az alábbiakban közöljük a Disk-Monitor program assembler-listáját, majd ezt követik a Commodore 64-hez és a VC 20-hoz alkalmazható, BASIC nyelven írt töltőprogramok.

P. 56.

```
symbol  value
 adr1    00fd     adr2    00fb     anzahl  0005     chkauf  aefa
 chkcom  aefd     chkstr  ad8f     chkzu   aef7     chrget  0073
 chrgot  0079     dscrpt  0064     execld  a7e7     execut  0308
 frestr  ba63     frmevl  ade9     getbyt  b79e     getvar  b08d
 gleich  00b2     ill     c866     ill1    c8cc     ilquan  b248
 integr  b1aa     laenge  0003     len1    0003     len2    0004
 loop    c908     loop1   c85b     mid     c883     midcod  00ca
 midstr  c86b     midtst  c876     next    c8c3     ok      c8db
 ok1     c8fb     poston  0004     pscode  00b9     setstr  aa52
 start   0006     store   c8cf     str     c849     str2    c850
 str3    c860     stradr  0062     strcde  00c4     string  c828
 syntax  af08     temp    0003     test    aeff     test2   c81e
 testin  c80d     tststr  ad8f     typflg  000c     varadr  0049
 varnam  0045     varstr  0005     vector  030a     yfac    b3a2
 zeig2   0050
 0000                ;      disk monitor vc20/cbm64
 0000                ;
 0000                prompt = $3c              ; <
 0000                ncomds = 6                ; utasitasok szama
 0000                input  = $ffcf
 0000                talk   = $ffd4
 0000                sctalk = $ff96
 0000                iecin  = $ffa5
 0000                untalk = $ffab
 0000                listen = $ffb1
 0000                sclist = $ff93
 0000                iecout = $ffa8
 0000                unlist = $ffae
 0000                write  = $ffd2
 0000                open   = $ffc0
 0000                close  = $ffc3
 0000                setpar = $ffba
 0000                setnam = $ffbd
 0000                chkin  = $ffc6
 0000                ckout  = $ffc9
 0000                clrch  = $ffcc
 0000                cr     = 13
 0000                quote  = $22
 0000                qtflg  = $d4
 0000                *      = $200             ; basic input puffer
 0200                savx   * = *+1
 0201                wrap   * = *+1
 0202                bad    * = *+1
 0203                von    * = *+1
 0204                bis    * = *+1
 0205                status = $90
 0205                sa     = $b9              ; masodlagos cim
 0205                fa     = $ba              ; egysegszam
 0205                fnadr  = $bb
```

```
0205                fnlen  = $b7
0205                tmpc   = $97
0205                count  = 8               ; byte-ok szama 8/sor
0205                ;                          vc-20-nal 4
0205                ;
0205                ready  = $e37b           ; vc-20-nal $e467
0205                ;
0205                       * = $c000
c000                ;
c000  a2 00         init   ldx #0            ; program fej
c002  bd 85 c2      msgout lda mesage,x
c005  20 d2 ff      jsr    write
c008  e8            inx
c009  e0 12         cpx    #ascdmp-mesage
c00b  d0 f5         bne    msgout
c00d  a2 0d         start  ldx #cr
c00f  a9 3c         lda    #prompt
c011  20 eb c0      jsr    wrtwo
c014  a9 00         lda    #0
c016  8d 01 02      sta    wrap
c019  20 33 c1      st1    jsr rdoc          ; input olvasas
c01c  c9 3c         cmp    #prompt
c01e  f0 f9         beq    st1
c020  c9 20         cmp    #$20              ; ' ' atugras
c022  f0 f5         beq    st1
c024  a2 05         s0     ldx #ncomds-1     ; osszehasonlitas az utasitastab
c026  dd 6a c0      s1     cmp comds,x
c029  d0 0c         bne    s2
c02b  8e 00 02      stx    savx
c02e  bd 70 c0      lda    adrh,x
c031  48            pha                      ; visszaugrasi cim a verembe
c032  bd 76 c0      lda    adrl,x
c035  48            pha
c036  60            rts
c037  ca            s2     dex
c038  10 ec         bpl    s1
c03a  4c 0d c0      jmp    start
c03d                ;
c03d                ;    blokktartalom kiiras a kepernyore
c03d                ;
c03d  85 97         dm     sta tmpc
c03f  20 62 c0      dm1    jsr space
c042  b9 e0 c2      lda    buffer,y          ; egy byte a pufferbol
c045  20 dc c0      jsr    wrob
c048  c8            iny
c049  d0 03         bne    dm2
c04b  ee 01 02      inc    wrap
c04e  c6 97         dm2    dec tmpc
c050  d0 ed         bne    dm1
c052  60            rts
c053                ;    egy byte olvasasa - tarcimnek
c053  20 fe c0      byt    jsr rdob
c056  90 03         bcc    by3               ; ures karakter ?
c058  99 e0 c2      sta    buffer,y          ; egy byte a pufferba
c05b  c8            by3    iny
c05c  c6 97         dec    tmpc
c05e  60            rts
c05f  20 62 c0      spac2  jsr space
c062  a9 20         space  lda #$20          ;'
c064  2c            .byte  $2c
c065  a9 0d         crlf   lda #cr
c067  4c d2 ff      jmp    write
c06a                ;
c06a                ;utasitas- es cim tablazat
c06a                ;
c06a  3a            comds  .byte $3a         ;':'  tartartalom atiras
c06b  57            .byte  $57               ;'w'  blokk irasa
```

```
c06c  52        .byte  $52            ;'r'  blokk olvasasa
c06d  4d        .byte  $4d            ;'m'  kiiras a kepernyore
c06e  40        .byte  $40            ;'@'  disk utasitas
c06f  58        .byte  $58            ;'x'  kilepes
c070  bf        adrh   .byte >altm-1
c071  c0        .byte  >direkt-1
c072  c0        .byte  >direkt-1
c073  bf        .byte  >dsplym-1
c074  c0        .byte  >disk-1
c075  e2        .byte  >ready-1
c076  c0        adrl   .byte <altm-1
c077  90        .byte  <direkt-1
c078  90        .byte  <direkt-1
c079  7b        .byte  <dsplym-1
c07a  3e        .byte  <disk-1
c07b  7a        .byte  <ready-1
c07c  a0 00     dsplym ldy #0
c07e  8c 03 02  sty    von
c081  88        dey
c082  8c 04 02  sty    bis
c085  20 cf ff  jsr    input
c088  c9 0d     cmp    #cr
c08a  f0 17     beq    dsp1
c08c  20 fe c0  jsr    rdob           ; startcim olvasas
c08f  90 12     bcc    dsp1
c091  8d 03 02  sta    von
c094  20 cf ff  jsr    input
c097  c9 0d     cmp    #cr
c099  f0 08     beq    dsp1
c09b  20 fe c0  jsr    rdob           ; vegcim olvasas
c09e  90 03     bcc    dsp1
c0a0  8d 04 02  sta    bis
c0a3  ac 03 02  dsp1   ldy von
c0a6  20 c6 c2  dsp2   jsr tstend
c0a9  20 d6 c2  jsr    altrit
c0ac  98        tya
c0ad  20 dc c0  jsr    wrob           ; cim
c0b0  20 62 c0  jsr    space          ; vc-20-nal kimarad
c0b3  a9 08     lda    #count         ; 8, vagy 4
c0b5  20 3d c0  jsr    dm             ; kiiratas
c0b8  20 97 c2  jsr    ascdmp         ;feltetlen ugras
c0bb  4c a6 c0  jmp    dsp2
c0be  4c 0d c0  beqs1  jmp start
c0c1            ; tartartalom atiras, cim es adat olvasas
c0c1  20 fe c0  altm   jsr rdob       ;cim olvasas
c0c4  90 f8     bcc    beqs1
c0c6  a8        tay
c0c7  a9 08     lda    #count         ;byte-ok szama
c0c9  85 97     sta    tmpc
c0cb  20 33 c1  jsr    rdoc           ; vc-20-nal kihagyni
c0ce  20 33 c1  a5     jsr rdoc
c0d1  20 53 c0  jsr    byt
c0d4  d0 f8     bne    a5
c0d6  20 97 c2  jsr    ascdmp
c0d9  4c 0d c0  jmp    start
c0dc            ;
c0dc            ;
c0dc  48        wrob   pha
c0dd  4a        lsr    a
c0de  4a        lsr    a
c0df  4a        lsr    a
c0e0  4a        lsr    a
c0e1  20 f4 c0  jsr    ascii          ; ascii kodra
c0e4  aa        tax
c0e5  68        pla
c0e6  29 0f     and    #%1111
c0e8  20 f4 c0  jsr    ascii
```

```
c0eb                ; karakter az x-be, a-ba
c0eb   48           wrtwo  pha
c0ec   8a           txa
c0ed   20 d2 ff     jsr    write
c0f0   68           pla
c0f1   4c d2 ff     jmp    write
c0f4   18           ascii  clc
c0f5   69 f6        adc    #$f6
c0f7   90 02        bcc    asci
c0f9   69 06        adc    #6
c0fb   69 3a        asci   adc #$3a
c0fd   60           rts
c0fe                ; hexbyte olvasasa a-ba
c0fe   a9 00        rdob   lda #0
c100   8d 02 02     sta    bad
c103   20 33 c1     jsr    rdoc
c106   c9 20        rdob1  cmp #$20          ; ' '
c108   d0 09        bne    rdob2
c10a   20 33 c1     jsr    rdoc              ; kovetkezo karakter olvasas
c10d   c9 20        cmp    #$20              ; ' '
c10f   d0 0f        bne    rdob3
c111   18           clc                      ; cy=0
c112                ;
c112   60           rts
c113   20 28 c1     rdob2  jsr hexit         ; konvertalo hex.re
c116   0a           asl    a
c117   0a           asl    a
c118   0a           asl    a
c119   0a           asl    a
c11a   8d 02 02     sta    bad
c11d   20 33 c1     jsr    rdoc
c120   20 28 c1     rdob3  jsr hexit
c123   0d 02 02     ora    bad
c126   38           sec                      ; cy=1
c127   60           rts
c128   c9 3a        hexit  cmp #$3a
c12a   08           php
c12b   29 0f        and    #%1111
c12d   28           plp
c12e   90 02        bcc    hex09             ; 0 - 9
c130   69 08        adc    #8                ; +9 (c-1)
c132   60           hex09  rts
c133   20 cf ff     rdoc   jsr input         ; karakter olvasas
c136   c9 0d        cmp    #cr               ; cr ?
c138   d0 f8        bne    hex09             ; nem, return
c13a   68           pla
c13b   68           pla                      ; igen, vissza a starthoz
c13c   4c 0d c0     jmp    start
c13f                ;
c13f                ; dos support
c13f   20 cf ff     disk   jsr input
c142   c9 0d        cmp    #cr
c144   d0 27        bne    dskcmd            ; disk parancs
c146   a9 00        lda    #0
c148   85 90        sta    status            ; st torles
c14a   20 65 c0     jsr    crlf
c14d   a9 08        lda    #8
c14f   85 ba        sta    fa                ; egysegszam
c151   20 d4 ff     jsr    talk
c154   a9 6f        lda    #15+$60
c156   85 b9        sta    sa                ; sa=15
c158   20 96 ff     jsr    sctalk            ; masodlagos cim
c15b   20 a5 ff     errin  jsr iecin
c15e   24 90        bit    status
c160   70 05        bvs    enddsk
c162   20 d2 ff     jsr    write
c165   d0 f4        bne    errin
```

```
c167  20 ab ff   enddsk jsr untalk
c16a  4c 0d c0   jmp    start
c16d  c9 24      dskcmd cmp #$24          ;'$'
c16f  f0 1d      beq    err1              ;catalog
c171  48         pha
c172  a9 08      lda    #8
c174  85 ba      sta    fa
c176  20 b1 ff   jsr    listen
c179  a9 6f      lda    #15+$60
c17b  85 b9      sta    sa
c17d  20 93 ff   jsr    sclist
c180  68         pla
c181  20 a8 ff   comdot jsr iecout
c184  20 cf ff   jsr    input
c187  c9 0d      cmp    #cr
c189  d0 f6      bne    comdot
c18b  20 ae ff   jsr    unlist
c18e  4c 0d c0   err1   jmp start
c191  20 33 c1   direkt jsr rdoc
c194  20 fe c0   jsr    rdob              ; track olvasas
c197  90 f5      bcc    err1
c199  8d 27 c2   sta    track
c19c  20 33 c1   jsr    rdoc
c19f  20 fe c0   jsr    rdob
c1a2  90 ea      bcc    err1
c1a4  8d 2a c2   sta    sector
c1a7  20 49 c2   jsr    opndir
c1aa  ad 00 02   lda    savx
c1ad  c9 01      cmp    #1
c1af  f0 1e      beq    drwrit
c1b1  a9 31      lda    #$31              ;'1'
c1b3  20 ed c1   jsr    secomd            ; 'block-read'
c1b6  a2 0d      ldx    #13
c1b8  20 c6 ff   jsr    chkin
c1bb  a2 00      ldx    #0
c1bd  20 cf ff   dirin  jsr input
c1c0  9d e0 c2   sta    buffer,x
c1c3  e8         inx
c1c4  d0 f7      bne    dirin
c1c6  20 cc ff   jsr    clrch
c1c9  20 6e c2   enddir jsr clsdir
c1cc  4c 0d c0   jmp    start
c1cf  20 2c c2   drwrit jsr bufpnt        ; puffer/pointer beallitas
c1d2  a2 0d      ldx    #13
c1d4  20 c9 ff   jsr    ckout
c1d7  a2 00      ldx    #0
c1d9  bd e0 c2   dirout lda buffer,x
c1dc  20 d2 ff   jsr    write
c1df  e8         inx
c1e0  d0 f7      bne    dirout
c1e2  20 cc ff   jsr    clrch
c1e5  a9 32      lda    #$32              ;'2'
c1e7  20 ed c1   jsr    secomd            ; 'block-write'
c1ea  4c c9 c1   jmp    enddir
c1ed  8d 20 c2   secomd sta comdst+1
c1f0  a2 0f      ldx    #15
c1f2  ad 27 c2   lda    track
c1f5  20 78 c2   jsr    nmbasc
c1f8  8e 27 c2   stx    track
c1fb  8d 28 c2   sta    track+1
c1fe  ad 2a c2   lda    sector
c201  20 78 c2   jsr    nmbasc
c204  8e 2a c2   stx    sector
c207  8d 2b c2   sta    sector+1
c20a  a2 0f      ldx    #15
c20c  20 c9 ff   jsr    ckout
c20f  a2 00      ldx    #0
```

```
c211   bd 1f c2    cmdot  lda comdst,x
c214   20 d2 ff    jsr    write
c217   e8          inx
c218   e0 0d       cpx    #bufpnt-comdst
c21a   d0 f5       bne    cmdot
c21c   4c cc ff    jmp    clrch
c21f   55 31       comdst .byte 'u1:13 0
c227   00          track  .byte 0,0,$20
c228   00
c229   20
c22a   00          sector .byte 0,0
c22b   00
c22c   a2 0f       bufpnt ldx #15
c22e   20 c9 ff    jsr    ckout
c231   a2 00       ldx    #0
c233   bd 41 c2    pntout lda buftxt,x
c236   20 d2 ff    jsr    write
c239   e8          inx
c23a   e0 08       cpx    #opndir-buftxt
c23c   d0 f5       bne    pntout
c23e   4c cc ff    jmp    clrch
c241   42 2d       buftxt .byte 'b-p 13 0'
c249   a9 0f       opndir lda #15
c24b   a8          tay
c24c   a2 08       ldx    #8
c24e   20 ba ff    jsr    setpar
c251   a9 00       lda    #0
c253   20 bd ff    jsr    setnam
c256   20 c0 ff    jsr    open
c259   a9 0d       lda    #13
c25b   a8          tay
c25c   a2 08       ldx    #8
c25e   20 ba ff    jsr    setpar
c261   a9 01       lda    #1
c263   a2 6d       ldx    #<dadr
c265   a0 c2       ldy    #>dadr
c267   20 bd ff    jsr    setnam
c26a   4c c0 ff    jmp    open
c26d   23          dadr   .byte '#'
c26e   a9 0d       clsdir lda #13
c270   20 c3 ff    jsr    close
c273   a9 0f       lda    #15
c275   4c c3 ff    jmp    close
c278   a2 30       nmbasc ldx #$30          ;'0'  hexa - ascii-re
c27a   38          sec
c27b   e9 0a       numb1  sbc #10
c27d   90 03       bcc    numb2
c27f   e8          inx
c280   b0 f9       bcs    numb1
c282   69 3a       numb2  adc #$39+1
c284   60          rts
c285   0d          mesage .byte cr
c286   44 49       .byte  'disk-monitor v1.0'
c297   a8          ascdmp tay
c298   38          sec
c299   e9 08       sbc    #count
c29b   a8          tay
c29c   20 62 c0    jsr    space
c29f   a9 12       lda    #18
c2a1   20 d2 ff    jsr    write
c2a4   a6 08       ldx    count
c2a6   b9 e0 c2    ac2    lda buffer,y
c2a9   29 7f       and    #$7f
c2ab   c9 20       cmp    #$20          ;'
c2ad   b0 04       bcs    ac3
c2af   a9 2e       lda    #$2e          ;'.
c2b1   d0 03       bne    ac4
```

```
c2b3   b9 e0 c2    ac3     lda buffer,y
c2b6   20 d2 ff    ac4     jsr write
c2b9   a9 00               lda    #0
c2bb   85 d4               sta    qtflg
c2bd   c8                  iny
c2be   ca                  dex
c2bf   d0 e5               bne    ac2
c2c1   a9 92               lda    #146
c2c3   4c d2 ff            jmp    write
c2c6   ad 01 02    tstend  lda wrap
c2c9   d0 06               bne    endend
c2cb   cc 04 02            cpy    bis
c2ce   b0 01               bcs    endend
c2d0   60                  rts
c2d1   68          endend  pla
c2d2   68                  pla
c2d3   4c 0d c0            jmp    start
c2d6   20 65 c0    altrit  jsr crlf
c2d9   a9 3a               lda    #$3a                    ;':'
c2db   a2 3c               ldx    #prompt
c2dd   4c eb c0            jmp    wrtwo
c2e0                       buffer = *
c2e0                       .end
```

```
symbol table

symbol value
 a5       c0ce    ac2      c2a6    ac3      c2b3    ac4      c2b6
 adrh     c070    adrl     c076    altm     c0c1    altrit   c2d6
 ascdmp   c297    asci     c0fb    ascii    c0f4    bad      0202
 beqs1    c0be    bis      0204    buffer   c2e0    bufpnt   c22c
 buftxt   c241    by3      c05b    byt      c053    chkin    ffc6
 ckout    ffc9    close    ffc3    clrch    ffcc    clsdir   c26e
 cmdot    c211    comdot   c181    comds    c06a    comdst   c21f
 count    0008    cr       000d    crlf     c065    dadr     c26d
 direkt   c191    dirin    c1bd    dirout   c1d9    disk     c13f
 dm       c03d    dm1      c03f    dm2      c04e    drwrit   c1cf
 dskcmd   c16d    dsp1     c0a3    dsp2     c0a6    dsplym   c07c
 enddir   c1c9    enddsk   c167    endend   c2d1    err1     c18e
 errin    c15b    fa       00ba    fnadr    00bb    fnlen    00b7
 hex09    c132    hexit    c128    iecin    ffa5    iecout   ffa8
 init     c000    input    ffcf    listen   ffb1    mesage   c285
 msgout   c002    ncomds   0006    nmbasc   c278    numb1    c27b
 numb2    c282    open     ffc0    opndir   c249    pntout   c233
 prompt   003c    qtflg    00d4    quote    0022    rdob     c0fe
 rdob1    c106    rdob2    c113    rdob3    c120    rdoc     c133
 ready    e37b    s0       c024    s1       c026    s2       c037
 sa       00b9    savx     0200    sclist   ff93    sctalk   ff96
 secomd   c1ed    sector   c22a    setnam   ffbd    setpar   ffba
 spac2    c05f    space    c062    st1      c019    start    c00d
 status   0090    talk     ffd4    tmpc     0097    track    c227
 tstend   c2c6    unlist   ffae    untalk   ffab    von      0203
 wrap     0201    write    ffd2    wrob     c0dc    wrtwo    c0eb
```

```
100 fori=49152 to 49887
110 read x:poke i,x: s=s+x :next
120 data162,0,189,133,194,32,210,255,232,224,18,208
130 data245,162,13,169,62,32,235,192,169,0,141,1
140 data2,32,51,193,201,62,240,249,201,32,240,245
150 data162,5,221,106,192,208,12,142,0,2,189,112
160 data192,72,189,118,192,72,96,202,16,236,76,13
170 data192,133,151,32,98,192,185,224,194,32,220,192
180 data200,208,3,238,1,2,198,151,208,237,96,32
190 data254,192,144,3,153,224,194,200,198,151,96,32
200 data98,192,169,32,44,169,13,76,210,255,58,87
```

```
210 data82,77,64,88,192,193,193,192,193,227,192,144
220 data144,123,62,122,160,0,140,3,2,136,140,4
230 data2,32,207,255,201,13,240,23,32,254,192,144
240 data18,141,3,2,32,207,255,201,13,240,8,32
250 data254,192,144,3,141,4,2,172,3,2,32,198
260 data194,32,214,194,152,32,220,192,32,98,192,169
270 data8,32,61,192,32,151,194,76,166,192,76,13
280 data192,32,254,192,144,248,168,169,8,133,151,32
290 data51,193,32,51,193,32,83,192,208,248,32,151
300 data194,76,13,192,72,74,74,74,74,32,244,192
310 data170,104,41,15,32,244,192,72,138,32,210,255
320 data104,76,210,255,24,105,246,144,2,105,6,105
330 data58,96,169,0,141,2,2,32,51,193,201,32
340 data208,9,32,51,193,201,32,208,15,24,96,32
350 data40,193,10,10,10,10,141,2,2,32,51,193
360 data32,40,193,13,2,2,56,96,201,58,8,41
370 data15,40,144,2,105,8,96,32,207,255,201,13
380 data208,248,104,104,76,13,192,32,207,255,201,13
390 data208,39,169,0,133,144,32,101,192,169,8,133
400 data186,32,180,255,169,111,133,185,32,150,255,32
410 data165,255,36,144,112,5,32,210,255,208,244,32
420 data171,255,76,13,192,201,36,240,29,72,169,8
430 data133,186,32,177,255,169,111,133,185,32,147,255
440 data104,32,168,255,32,207,255,201,13,208,246,32
450 data174,255,76,13,192,32,51,193,32,254,192,144
460 data245,141,39,194,32,51,193,32,254,192,144,234
470 data141,42,194,32,73,194,173,0,2,201,1,240
480 data30,169,49,32,237,193,162,13,32,198,255,162
490 data0,32,207,255,157,224,194,232,208,247,32,204
500 data255,32,110,194,76,13,192,32,44,194,162,13
510 data32,201,255,162,0,189,224,194,32,210,255,232
520 data208,247,32,204,255,169,50,32,237,193,76,201
530 data193,141,32,194,162,15,173,39,194,32,120,194
540 data142,39,194,141,40,194,173,42,194,32,120,194
550 data142,42,194,141,43,194,162,15,32,201,255,162
560 data0,189,31,194,32,210,255,232,224,13,208,245
570 data76,204,255,85,49,58,49,51,32,48,32,0
580 data0,32,0,0,162,15,32,201,255,162,0,189
590 data65,194,32,210,255,232,224,8,208,245,76,204
600 data255,66,45,80,32,49,51,32,48,169,15,168
610 data162,8,32,186,255,169,0,32,189,255,32,192
620 data255,169,13,168,162,8,32,186,255,169,1,162
630 data109,160,194,32,189,255,76,192,255,35,169,13
640 data32,195,255,169,15,76,195,255,162,48,56,233
650 data10,144,3,232,176,249,105,58,96,13,68,73
660 data83,75,45,77,79,78,73,84,79,82,32,86
670 data49,46,48,152,56,233,8,168,32,98,192,169
680 data18,32,210,255,162,8,185,224,194,41,127,201
690 data32,176,4,169,46,208,3,185,224,194,32,210
700 data255,169,0,133,212,200,202,208,229,169,146,76
710 data210,255,173,1,2,208,6,204,4,2,176,1
720 data96,104,104,76,13,192,32,101,192,169,58,162
730 data62,76,235,192
740 ifs<>90444 then print "hiba a data-sorokban levo adatoknal!":end
750 sys49152
```

```
100 poke55,6690 and 255: poke56,6690/256: clr
110 for i =6690to7056
120 readq:pokei,q:s=s+q:next
130 data 162,0,189,164,28,32,210,255,232,224,18,208,245,162,13,169
140 data 62,32,7,27,169,0,141,1,2,32,79,27,201,62,240,249
150 data 201,32,240,245,162,5,221,140,26,208,12,142,0,2,189,146
160 data 26,72,189,152,26,72,96,202,16,236,76,47,26,133,151,32
170 data 132,26,185,0,29,32,248,26,200,208,3,238,1,2,198,151
180 data 208,237,96,32,26,27,144,3,153,0,29,200,198,151,96,32
190 data 132,26,169,32,44,169,13,76,210,255,58,87,82,77,64,88
```

```
200 data 26,27,27,26,27,228,223,175,175,157,90,102,160,0,140,3
210 data 2,136,140,4,2,32,207,255,201,13,240,23,32,26,27,144
220 data 18,141,3,2,32,207,255,201,13,240,8,32,26,27,144,3
230 data 141,4,2,172,3,2,32,229,28,32,245,28,152,32,248,26
240 data 169,4,32,95,26,32,182,28,76,200,26,76,47,26,32,26
250 data 27,144,248,168,169,4,133,151,32,79,27,32,117,26,208,248
260 data 32,182,28,76,47,26,72,74,74,74,74,32,16,27,170,104
270 data 41,15,32,16,27,72,138,32,210,255,104,76,210,255,24,105
280 data 246,144,2,105,6,105,58,96,169,0,141,2,2,32,79,27
290 data 201,32,208,9,32,79,27,201,32,208,15,24,96,32,68,27
300 data 10,10,10,10,141,2,2,32,79,27,32,68,27,13,2,2
310 data 56,96,201,58,8,41,15,40,144,2,105,8,96,32,207,255
320 data 201,13,208,248,104,104,76,47,26,32,207,255,201,13,208,39
330 data 169,0,133,144,32,135,26,169,8,133,186,32,180,255,169,111
340 data 133,185,32,150,255,32,165,255,36,144,112,5,32,210,255,208
350 data 244,32,171,255,76,47,26,201,36,240,29,72,169,8,133
360 if s<>35614 then print"hiba a data-sorokban levo adatoknal":end
370 load"p56/4",8: rem masodik resz behivasa
```

```
100 clr:for i =7057to7422
110 readq:pokei,q:s=s+q:next
120 data 186,32,177,255,169,111,133,185,32,147,255,104,32,168,255,32
130 data 207,255,201,13,208,246,32,174,255,76,47,26,76,47,26,32
140 data 79,27,32,26,27,144,245,141,70,28,32,79,27,32,26,27
150 data 144,234,141,73,28,32,104,28,173,0,2,201,1,240,30,169
160 data 49,32,12,28,162,13,32,198,255,162,0,32,207,255,157,0
170 data 29,232,208,247,32,204,255,32,141,28,76,47,26,32,75,28
180 data 162,13,32,201,255,162,0,189,0,29,32,210,255,232,208,247
190 data 32,204,255,169,50,32,12,28,76,232,27,141,63,28,162,15
200 data 173,70,28,32,151,28,142,70,28,141,71,28,173,73,28,32
210 data 151,28,142,73,28,141,74,28,162,15,32,201,255,162,0,189
220 data 62,28,32,210,255,232,224,13,208,245,76,204,255,85,49,58
230 data 49,51,32,48,32,0,0,32,0,0,162,15,32,201,255,162
240 data 0,189,96,28,32,210,255,232,224,8,208,245,76,204,255,66
250 data 45,80,32,49,51,32,48,169,15,168,162,8,32,186,255,169
260 data 0,32,189,255,32,192,255,169,13,168,162,8,32,186,255,169
270 data 1,162,140,160,28,32,189,255,76,192,255,35,169,13,32,195
280 data 255,169,15,76,195,255,162,48,56,233,10,144,3,232,176,249
290 data 105,58,96,13,68,73,83,75,45,77,79,78,73,84,79,82
300 data 32,86,49,46,48,152,56,233,4,168,32,132,26,169,18,32
310 data 210,255,162,4,185,0,29,41,127,201,32,176,4,169,46,208
320 data 3,185,0,29,32,210,255,169,0,133,212,200,202,208,229,169
330 data 146,76,210,255,173,1,2,208,6,204,4,2,176,1,96,104
340 data 104,76,47,26,32,135,26,169,58,162,62,76,7,27
350 ifs<>39469 then print"hiba a data-sorokban levo adatoknal":end
360 sys6690
```

5. FEJEZET
A NAGY CBM-LEMEZEGYSÉGEK

5.1. Az IEC busz és a soros busz

A Commodore 64-es és VC 20-as gépekhez egy beépített soros buszon keresztül csatlakoztathatjuk a különböző perifériákat, mint pl. a VC 1541-es lemezegységet, a nyomtatót vagy a rajzgépet.
A soros busz működési elve lehetővé teszi egyszerre több külső egység csatlakoztatását. Ahhoz, hogy működés közben egyértelműen hivatkozhassunk az egyes egységekre, az egységeket azonosító számokkal kellett ellátni. A lemezegység hivatkozási száma 8, a nyomtatóé 4.
Az egységhez a hivatkozási számmal hozzárendelünk egy ún. *elsődleges cím*et az OPEN utasításban.
Pl. az

```
OPEN 1,4
```

utasítással a nyomtatóhoz hozzárendeltük az 1-es elsődleges címet, azaz egy csatornát.
Mivel egyszerre több lemezegységgel is dolgozhatunk, ezek további megkülönböztetésére szolgál az ugyancsak az OPEN utasításban megadott másodlagos cím.
16 másodlagos címet használhatunk, Ø–15-ig számozva, azonban ezek közül háromnak különleges szerepe van:

Ø. – a programok betöltése
1. – a programok tárolása
15. – a parancs- és hibacsatorna

A további 13-at (2–14-ig) szabadon használhatjuk adathalmazok kezelésére.

A Commodore 64-es és a VC 1541-es egység közötti adatátvitel a soros buszon történik. A „soros" szó arra utal, hogy az információ bitenként, egyetlen vezetéken továbbítódik. Az átvitt információt azonban az alapgép is és a lemezegység is byte-onként (8 bit) dolgozza fel. Ahhoz, hogy a küldő és a fogadó egység között az adatforgalom során összhang legyen, szükség van egy további, ún. „handshake" (kézfogás) vezetékre.
Az adatátvitel lebonyolításában hat vonal vesz részt:

PONT	MEGJELÖLÉS
1	SRQ IN
2	föld
3	ATN
4	CLOCK
5	DATA
6	RESET

Ha a gép adatokat kíván közölni az egységgel, az ATN vezeték 1-re állítódik. A jelet észlelve a soros buszon levő minden egység felfüggeszti a pillanatnyi munkáját és átveszi az érkező byte-ot. Az adatok bitenként érkeznek a DATA vonalon. Hogy a fogadó egységhez mikor

érkezik a következő bit, azt az ún. CLOCK (óra, ütem) vonal jelzi. Ha az itt átvitt byte-ban meghatározott egységszám nem azonos az adott egység számával, az egység figyelmen kívül hagyja a küldött adatokat.
A címzett egység fogadja a következő adatot, nevezetesen az ún. *másodlagos címet*. Az egységszámmal együtt a byte maradék három bitje azt is meghatározza, hogy a kiválasztott egység adatokat fogadjon (LISTEN) vagy küldjön (TALK). Az utóbbi három bit értékétől függően megindul az adatáramlás a gép felé, vagy a géptől az egységhez.
A RESET vonal bekapcsolás után, a soros buszon csatlakoztatott minden egységet alapállapotba hozat. Az SRQ IN vonalon (Service Reqest) keresztül jelentik készültségi állapotukat az egyes egységek, pl. hogy az adatok az átvitelre készen állnak (ezt a vezetéket az alapgép operációs rendszere nem kérdezi le).
Több lemezegység alapállapotban nem használható, hacsak nincsenek fizikailag átcímezve különböző egységszámokra. A DISK ADR CHANGE programmal „szoftveresen" átcímezhetjük az egységeket, de természetesen kikapcsolás után az új egységszámok megszűnnek. Végleges átcímzést csak fizikailag, forrasztással lehet elvégezni.
A soros busz elvéhez hasonló elven működnek az IEC- és az IEEE 488 buszok. A legfontosabb eltérés a soros buszhoz képest az, hogy a fentieken az adatok nem sorosan, hanem párhuzamosan, egyszerre 8 adatvonalon áramlanak. Mivel még egy további „handshake" vonal is szükséges, az IEC busz egy összesen 24 eres kábellel dolgozik. Az IEEE 488-as busz legfőbb előnye a soros buszhoz képest a gyorsaság, azaz a 8 bit egyidejű átvitele. Az IEC busz kb. ötször gyorsabb a soros busznál: átviteli sebessége 1,8 kbyte, míg a soros buszé 0,4 kbyte. Egy 10 kbyte-os program betöltése a VC 1541-en kb. 25 másodpercig tart, míg a nagy CBM 2031-es készüléken kevesebb mint 6 másodpercig.
A sebességkülönbség mindenképpen indokolja, hogy az alapgéphez egy IEC buszt vásároljunk! Az IEC busz bővítéssel az alapgéphez minden nagy CBM egységet csatlakoztathatunk.

5.2. A CBM lemezegységek összehasonlítása

Az alábbi táblázat összefoglalja az összes CBM lemezegység műszaki adatait.

TÍPUS	1541	2031	4040	8050	8250
DOS-változat(ok)	2.6	2.6	2.1/2.7	2.5/2.7	2.7
Meghajtóegység	1	1	2	2	2
Meghajtóegységenkénti fejek száma	1	1	1	1	2
Tárkapacitás	170K	170K	340K	1.05M	2.12M
Soros file	168K	168K	168K	521K	1.05M
Relatív file	167K	167K	167K	183K/518K	1.04M
Puffertár (kbyte)	2	2	4	4	4
Sávok	35	35	35	77	77
Sávonkénti szektorok	17–21	17–21	17–21	23–29	23–29
Blokkonkénti byte-ok	256	256	256	256	256
Szabad blokkok	664	664	1328	4104	8266
Tartalomjegyzék és BAM sáv	18	18	18	38/39	38/39
File-bejegyzések	144	144	144	224	224
Átviteli sebesség (kbyte/s)	40	40	40	40	40
Belső soros IEC buszon keresztül	0.4	1.8	1.8	1.8	1.8

Elérési idők (m/s):	30	30	30	5	5
sávról sávra					
átlagos idő	360	360	360	125	125
Percenkénti fordulatszám	300	300	300	300	300

A nagy CBM lemezegységek áttekintése

A VC 1541-es lemezegység a legkisebb tárkapacitású CBM-egység, de mindezideig az egyetlen készülék, amely soros buszon keresztül közvetlenül kapcsolódik a Commodore 64-hez és a VC 20-hoz. A funkciókat, a szerkezeti felépítést és a működési elvet tekintve azonos a CBM 2031-es lemezegységgel. A CBM 2031-es a VC 1541-hez viszonyítva csak abban mutat eltérést, hogy soros busz helyett párhuzamos IEEE 488 busszal van ellátva. Ez az eltérés mintegy ötszörösére növeli az átviteli sebességet. Ha a CBM 2031-et a Commodore 64-hez vagy a VC20-hoz akarjuk csatlakoztatni, egy IEC busz-modult kell vásárolni. A CBM 2031-es tárformátuma teljesen kompatibilis a VC 1541-es tárformátumával. Mindkét készüléknél lemezenként 170 kbyte áll rendelkezésre. A CBM 2031-es által felírt információt a VC 1541-es tudja olvasni, felülírni és megfordítva. Ez érvényes a sorozat következő típusára, a CBM 4040-es lemezegységre is. A 4040-es kettős meghajtóegység, kétszer 170 kbyte-os tárkapacitással.
A kettős meghajtóegység előnye nemcsak a kétszeres tárkapacitás, hanem az, hogy az adatok egyik meghajtóegységből átvihetők a másikba. A programok és file-ok átmásolását a COPY utasítás segítségével valósíthatjuk meg, például az

```
OPEN 1,8,15, "CL:TEST=TEST"      vagy a
COPY "TEST", D0 TO "TEST", D1
```

a TEST file-t azonos névvel átmásolja a 0. meghajtóegységről az 1-re. Különböző meghajtóegységekről több file-t is összekapcsolhatunk a CONCAT utasítással.
A kettős meghajtóegység legfontosabb előnye azonban mégis a teljes lemezek másolási lehetősége. A lemezmásolást a fentihez hasonlóan szintén egy utasítással elvégezhetjük, melynek hatására a lemezegység automatikusan megformálja az új lemezt, majd a másolandó lemez tartalmát sávról-sávra átmásolja a megformált lemezre. A lemezmásolás utasítása:

```
OPEN 1,8,15,"D1=0"      vagy
BACKUP D0 TO D1.
```

A 4040-es készüléken a másolás kevesebb mint 3 percet vesz igénybe; közben az alapgép tovább dolgozhat, mivel a lemezegység operációs rendszere végzi ezt a műveletet.
A CBM 8050-es és 8250-es egységek kétszeres sűrűséggel ('double density', 77 sáv) írják a lemezre az információt, következésképpen a 4040-es egységek nem kompatibilisek a 8050/8250 egységekkel, azaz kölcsönösen nem tudják olvasni a másik által teleírt lemezeket. Vannak azonban olyan programok (pl. a COPY ALL program), amelyekkel átmásolhatjuk az 1541/4040-es egységek által felírt programokat és adatokat a 8050/8250-es lemezegységekre. A 8050-es és a 8250-es egységeken a dupla írássűrűség miatt jóval több adat fér el, mint a kis egységeken: a 8050-es egységen 1 Mbyte, a 8250-es egységen pedig 2 Mbyte. A 8250-es kétszeres kapacitása abból adódik, hogy a lemezegység a lemez mindkét oldalát tudja használni ('double sided'), mivel meghajtóegységenként 2-2 író/olvasófejjel rendelkezik. Ahhoz, hogy relatív file esetén is kihasználhassuk a teljes kapacitást, az operációs rendszer ún. Super-Side-Sector-ral dolgozik, amely az egyenként 6 oldalszektor-blokkból álló

127 csoport mutatóit tartalmazza. Ezzel az adatszervezéssel egy relatív file (elméletileg) 23 Mbyte-ra is kiterjedhet (a 8050-esnél a 2.7 DOS-változattól kezdődően). Az IEC buszon keresztül bármelyik lemezegység csatlakoztatható a Commodore 64-hez és a VC20-hoz, s így "on line" üzemben ezekkel a gépekkel is több megabyte-nyi adathalmazt dolgozhatunk fel.
A nagy CBM egységek további előnye, hogy puffertáruk kétszer akkora, mint a kis egységeké.
A nagyobb puffertár felhasználásával egyidejűleg öt soros file, vagy legfeljebb három relatív file lehet megnyitott állapotban (vagy ami ezzel egyenértékű, egyszerre két relatív és két soros file).

A következő részben összehasonlítjuk az 1541/4040-es lemezformátumot a 8050/8250-es formátummal, rámutatunk a BAM és a tartalomjegyzék elhelyezkedésének és felépítésének eltéréseire.
A 8050/8250-es formátumnál a BAM és a tartalomjegyzék a 38. és a 39. sávot használja. A 39. sáv Ø. szektorában helyezkedik el a lemez neve és a formátum ismertetőjele.

```
>:ØØ  26  ØØ  43  ØØ  ØØ  ØØ  43  42  &.C...CB
>:Ø8  4E  2Ø  38  3Ø  35  3Ø  AØ  AØ  M   8050
>:10  AØ  AØ  AØ  AØ  AØ  AØ  AØ  AØ
>:18  30  31  AØ  32  43  AØ  AØ  AØ  Ø1  2C
```

A Ø. és az 1. byte tartalmazza az első BAM-blokk sáv szektormutatóját (38. sáv, Ø. szektor), a 2. byte pedig a formátum "C" jelölő betűjét. A 3-tól 5-ig terjedő byte-ok használaton kívül vannak. A 6-tól 31-ig terjedő byte-okban "Shift Space"-szel van kitöltve a lemez neve, a mi esetünkben a "CBM 8050". A 24. és a 25. byte tartalmazza a 'Ø1' lemezazonosítót (ID-t), a 27. és a 28. pedig a '2C' DOS-formátumot.
A blokk fennmaradó része kihasználatlan.
A BAM nem fér el egy blokkban, így a 38. sávon el van osztva; a 8050-nél a Ø. és a 3. szektorban, a 8250-nél pedig ezen kívül 6. és 9. szektorokban is. Mivel egy sávon több szektor van, az egyes sávokra vonatkozó BAM-bejegyzést bővíteni kellett, a bejegyzés 5 byte-ot foglal le. Az első byte itt is a sávonkénti szabad szektorok számát tartalmazza, a következő byte-ok pedig a szabad és foglalt szektorok bittérképét tartalmazzák (Ø= a szektor foglalt, 1= a szektor szabad).
Részletesen bemutatjuk egy lemez 38. sáv Ø. szektorának tárkivonatát:

```
>:ØØ  26  Ø3  43  ØØ  Ø1  33  1D  FF
>:Ø8  FF  FF  1F  1D  FF  FF  FF  1F
>:1Ø  1D  FF  FF  FF  1F  1D  FF  FF
>:18  FF  1F  1D  FF  FF  FF  1F  1D
>:2Ø  FF  FF  FF  1F  1D  FF  FF  FF
>:28  1F  1D  FF  FF  FF  1F  1D  FF
>:3Ø  FF  FF  1F  1D  FF  FF  FF  FF
>:38  1D  FF  FF  FF  1F  1D  FF  FF
>:4Ø  FF  1F  1D  FF  FF  FF  1F  1D
>:48  FF  FF  FF  1F  1D  FF  FF  FF
>:5Ø  1F  1D  FF  FF  FF  1F  1D  FF
>:58  FF  FF  1F  1D  FF  FF  FF  1F
>:6Ø  1D  FF  FF  FF  1F  1D  FF  FF
>:68  FF  1F  1D  FF  FF  FF  1F  1D
>:7Ø  FF  FF  FF  1F  1D  FF  FF  FF
```

```
>:78  1F  1D  FF  FF  FF  1F  1D  FF
>:80  FF  FF  1F  1D  FF  FF  FF  1F
>:88  1D  FF  FF  FF  1F  1D  FF  FF
>:90  FF  1F  1D  FF  FF  FF  1F  1D
>:98  FF  FF  FF  1F  1D  FF  FF  FF
>:A0  1F  1D  FF  FF  FF  1F  1D  FF
>:A8  FF  FF  1F  18  FC  F3  EF  1F
>:B0  00  00  00  00  00  00  00  00
>:B8  00  00  00  00  00  00  00  0F
>:C0  F4  93  46  1A  18  6C  FB  FF
>:C8  1F  00  00  00  00  00  00  00
>:D0  00  00  00  00  00  00  00  00
>:D8  05  00  00  4D  04  1B  FF  FF
>:E0  FF  07  1B  FF  FF  FF  07  1B
>:E8  FF  FF  FF  07  1B  FF  FF  FF
>:F0  07  1B  FF  FF  FF  07  1B  FF
>:F8  FF  FF  07  1B  FF  FF  FF  07
```

A 0. és az 1. byte itt is a következő BAM-blokkra utal, ezen a lemezen a 38. sáv 3. szektorára. A 2. byte a formátumjelölő "C" betűt tartalmazza. A 4. byte-ban helyezkednek el azok a sávszámok, amelyre az adott BAM-rész vonatkozik (itt az 1-től 51-ig terjedő sávok). A 6. pozíciótól kezdődően találjuk az egyes sávokra vonatkozó 5 byte-os bejegyzéseket. A következő BAM-blokk is hasonló felépítésű, a 8050-esnél az 52-től 77-ig terjedő sávokra vonatkozik és legfeljebb 140 byte-ot foglal le. Az utolsó BAM-blokk minden esetben az első tartalomjegyzék blokkra utal: a 39. sáv 1. szektorára.

A 8250-esnél a BAM-hoz 4 blokk szükséges. A 38. sáv 0. szektora tartalmazza az 1-től 51-ig terjedő sávokra, a 38. sáv 3. szektora az 52-től 100-ig terjedő sávokra, majd végül a 38. sáv 9. szektora a 151-től 154-ig terjedő sávokra vonatkozó háttértérképet.
A tartalomjegyzék sáv, vagyis a 39. sáv még további 28 szabad blokkot tartalmaz. Így tehát 28* 8-224 tartalomjegyzék-bejegyzés lehetséges, ellentétben az 1541/4040-nel, ahol a tartalomjegyzék-bejegyzések száma legfeljebb 144 lehet. A tartalomjegyzék felépítése az összes formátumnál azonos.

Az alábbi táblázat szemléletesen mutatja be a különböző típusú egységek sáv-szektor felépítését:

	1541/4040	8050/8250	
Sávok	1 –17 : 0–20	1 –39 : 0–28	szektorok
	18–24 : 0–18	40–53 : 0–26	
	25–24 : 0–17	54–64 : 0–24	
	31–35 : 0–16	65–77 : 0–22	
		csak 8250:	
		78–116 : 0–28	
		117–130 : 0–26	
		131–141 : 0–24	
		142–154 : 0–22	
Blokkok	683	2083/4186	
Szabad blokkok	664	2052/4133	

TOVÁBBI DATA BECKER KÖNYVEK A NOVOTRADE RT. KIADÁSÁBAN

A Commodore 64 belső felépítése

A könyv belülről világítja meg az Ön kedvenc számítógépét. Nélkülözhetetlen mindazok számára, akik a mikrogép rejtelmeinek mélyére akarnak ásni. A több mint 300 oldalas kötet a C 64-es minden szempontból alapos leírását adja. Teljes ROM listát tartalmaz részletes magyarázatokkal felszerelve, így Ön kedve szerint tanulmányozhatja a BASIC interpreter, a kernal vagy operációs rendszer működését. Számos példát talál majd a gépi nyelvű programozásra és továbbiakat, melyek révén a programozó munkája kellemesebb lesz.

Tippek és trükkök a Commodore 64-esen

A kötet könnyedén elsajátítható programozási technikák és fogások gyűjteménye. Az egyes fejezetek többek közt grafikus programokat, kényelmes adatbevitelt, magas szintű BASIC ismertetést, a CP/M alkalmazhatóságát és számos gyakorlati megoldást tartalmaznak. Szellemes ötleteket kapnak Olvasóink válogatási feladatokhoz, a változók kezeléséhez, a POKE-ok használatához. Mindent egybevetve, a kötet a programozás praktikus eszköztárának gyűjteménye.

Gépi kódú programozás a Commodore 64-esen

Azok számára íródott akik már nem a BASIC nyelven, hanem a gyorsabb, a memóriát hatékonyabban kihasználó gépi kódban kívánnak programozni. A könyv kifejezetten a C-64-eshez íródott. Tanulja meg, hogyan működik a 6510-es mikroprocesszor a C-64-esen. A ROM rutinok hozzáférhetősége, I/O, BASIC bővítés stb. Tartalmazza 3 teljes program listáját: egy ASSEMBLER-ét, egy DISASSEMBLER-ét és egy nem mindennapi 6510 SIMULATOR-ét, mely valóban "láttatja" Önnel a C-64-es működését.

Ára: 355,- Ft